PNL-6425
UC-70

2F

User's Guide for West Valley Feed Preparation Code (Recipe)

D. W. Faletti
T. A. Erb
H. Harty
C. A. LoPresti

February 1988

Prepared for the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Operated for the U.S. Department of Energy
by Battelle Memorial Institute

Battelle

PNL-6425

DISCLAIMER

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE MEMORIAL INSTITUTE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC06-76RLO 1830

Printed in the United States of America
Available from
National Technical Information Service
United States Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161

NTIS Price Codes
Microfiche A01

Printed Copy

Pages	Price Codes
001-025	A02
026-050	A03
051-075	A04
076-100	A05
101-125	A06
126-150	A07
151-175	A08
176-200	A09
201-225	A010
226-250	A011
251-275	A012
276-300	A013

PNL-6425
UC-70

USER'S GUIDE FOR WEST VALLEY FEED PREPARATION CODE (RECIPE)

D. W. Faletti
T. A. Erb
H. Harty
C. A. LoPresti

February 1988

Prepared for
the U.S. Department of Energy
under Contract DE-AC06-76RL0 1830

Pacific Northwest Laboratory
Richland, Washington 99352

ACKNOWLEDGMENTS

The authors acknowledge the assistance of the following Pacific Northwest Laboratory personnel: researcher Barry Wise, who designed the matrix of additive weight percents, and Suzanne Liebetrau and DeDe Johnson for their help in the preparation and production of the manuscript.

SUMMARY

The RECIPE code was developed by Pacific Northwest Laboratory under the West Valley Support Task of the Nuclear Waste Treatment Program. The West Valley Demonstration Project is constructing a waste vitrification facility to immobilize high-level waste at the West Valley site. Additional chemicals must be added to the high-level wastes to produce an acceptable glass. In order to maintain the composition of the glass within the targeted compositional region, the chemical additions must be carefully controlled. The amounts and kinds of chemicals that will be required will vary from waste batch to waste batch because the composition and amount of waste will vary among batches, and because the composition of the additives will vary among additive batches.

The RECIPE code was developed for two purposes:

- to enable accurate computation of the amounts of additives required for a given batch of waste
- to determine whether a batch of slurry will indeed give an acceptable glass composition when mixed with the batch of waste created from the additives.

In addition, RECIPE can conduct computations necessary for determining the amount and type of additives required to return an off-specification batch of feed or the melter contents to tolerance levels.

This report provides the user with a) an overview of the glass-making process and RECIPE's role in it, and b) sufficient information that, when combined with the extensive documentation within the source code, the RECIPE code can be used, maintained, and modified.

CONTENTS

ACKNOWLEDGMENTS iii

SUMMARY v

1.0 INTRODUCTION 1.1

2.0 BACKGROUND 2.1

PROCESS DESCRIPTION 2.1

GLASS AND FEED COMPOSITIONS 2.4

3.0 MATHEMATICAL SOLUTION 3.1

EQUATIONS 3.1

COMPUTATION OF SHIMS FOR THE MFT AND FOR THE MELTER 3.6

COMPUTATION OF CORRECTIVE ADDITIONS TO THE SLURRY IN THE CCMT OR SHIM PREPARATION TANK 3.6

OTHER COMPUTATIONS 3.7

4.0 IMPLEMENTATION GUIDE AND INPUT DESCRIPTION 4.1

PART A: RECIPE REQUIREMENTS 4.1

PART B: RECIPE INPUT FILE FORMATS 4.1

5.0 OPERATION OF RECIPE 5.1

EXAMPLE 1 . . . . . 5.1

EXAMPLE 2 . . . . . 5.24

6.0 TEST CASES 6.1

7.0 LIST OF RECIPE MODULES 7.1

MODULE DESCRIPTIONS 7.1

LIST OF MODULES IN ORDER OF CALLING 7.11

1.0 INTRODUCTION

The RECIPE code was developed by Pacific Northwest Laboratory (PNL) under the West Valley Support Task of the Nuclear Waste Treatment Program. The West Valley Demonstration Project (WVDP) is constructing a waste vitrification system that utilizes a slurry-fed ceramic melter (SFCM) to immobilize high-level wastes at the West Valley Site. During the vitrification process, composition of the glass can be held acceptably close to a target composition by adding the proper amounts of chemicals to the waste during the glass-batching process. The amounts and kinds of chemicals that are required vary from waste batch to waste batch because the composition and amount of waste vary from batch to batch, and because the composition of the additives varies from additive batch to additive batch.

The RECIPE code was developed with a dual purpose:

- to enable an accurate computation of the amounts of additives required for a given batch of waste
- to determine whether a batch of slurry created from the additives will indeed result in an acceptable glass composition when mixed with the batch of waste.

In addition, RECIPE can conduct computations necessary for determining the amount and type of additives required to return an off-specification batch of feed or the melter contents to tolerance levels.

This document was developed with the following goals:

- give the user an overview of the glass-batching process
- describe RECIPE's role in that process
- provide explicit instructions for using the code
- provide enough information about each module's function so that the user can modify the existing code if so desired
- provide test cases with documented output.

2.0 BACKGROUND

RECIPE's function is to determine the amounts of chemicals that should be added 1) to the concentrator feed makeup tank (CFMT) or the melter feed tank (MFT) to bring the feed composition to the SFCM within specifications, or 2) to the melter to bring the composition of the glass within specifications. RECIPE will not be used to control the solids loading, i.e. the amount of water in the feed. Instead it will compute the amount of glass-forming additives that, when added to the tank, will give a composition (expressed in terms of glass-forming oxides) such that the ratios between the concentrations of the glass-forming oxides in the tank match the ratios between the concentrations of the glass-forming oxides of the desired feed.

At this point, it will be helpful to describe the relevant processes and define some basic terms.

PROCESS DESCRIPTION

A schematic of the vitrification process is given in Figure 1. Shown are the CFMT and the cold-chemical makeup tank (CCMT), which are used to prepare the feed. The batches of feed are sent to the melter feed tank (MFT). The feed then continuously flows to the melter, where the glass is produced. Water and volatile solids exit to the off-gas system; the glass is poured into steel canisters.

For the purposes of this document, feed preparation can be considered to begin in the CFMT. The waste batch is concentrated and homogenized in the CFMT, and the material is then sampled. Concurrent with this sampling, an accurate measurement is taken of the volume and density of the material in the CFMT so that an accurate measurement of the mass of each oxide of interest can be obtained. RECIPE will use this information to determine the amount of additives required to meet the desired feed composition.

The required additives, along with water, are added to the CCMT to form a slurry. This slurry will be homogenized and subjected to chemical analysis. The RECIPE code, in combination with the known composition and

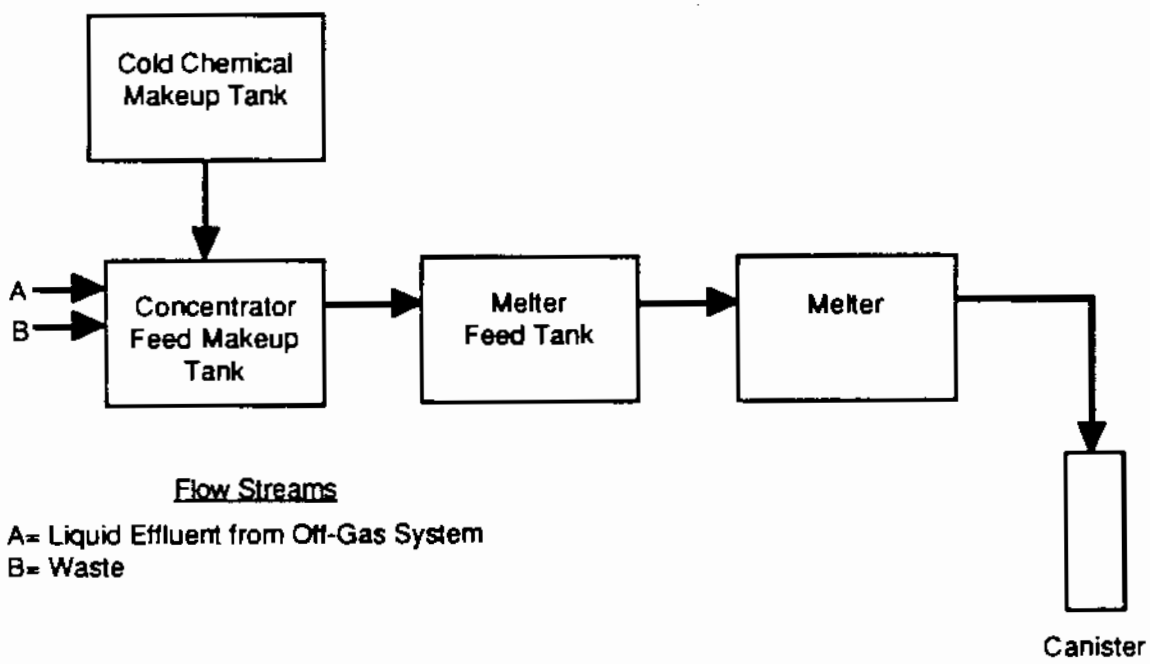

FIGURE 1. Schematic of WVDP Vitrification Process

contents of the CFMT and CCMT, will be used to determine whether the proper amount of chemicals have been added to the CCMT. If the chemical composition is not correct, an adjustment (shim) will be computed by RECIPE, and additional chemicals (if the amounts required are practical) can be added to the CCMT.

Once the CCMT concentration is judged to be correct, the contents are added to the CFMT. Following homogenization, a batch of feed is transferred from the CFMT to the MFT, leaving a heel in the CFMT. The existence of this heel is of no concern to the use of RECIPE; it becomes part of the next batch of waste.

Once the newly added batch of feed has been thoroughly mixed with the heel left in the MFT[(a)], a sample is taken of the MFT contents. If the results of the chemical analysis indicate that the composition of the material in the MFT is outside of specifications, RECIPE can be used to compute the additives necessary to make a shim that will bring the MFT composition back to tolerance. If practical, this shim could be added to the MFT to bring its contents within compositional tolerances.

The authors do not know what tank or tanks will be used to prepare shims. The CCMT may not be appropriate because the quantity required for practical shims will most likely be small compared with the original addition to the CFMT. Details about the tank used to make up a slurry are unnecesary for use of RECIPE. However, RECIPE operates under the assumption that there are no heels in any of the tanks used to prepare material to be added to the CFMT, MFT, or melter.

The procedure described above should result in feeds of correct composition. If it is successful, the contents of the melter will remain within acceptable tolerances. However, it is possible that feed composition may be out of tolerance long enough that the composition of the melter contents also become out of tolerance. For example, this may happen if the feed composition is in error between the time when the sample of the MFT is taken and chemical analysis is received. If the melter composition is out of tolerance, West Valley Nuclear Services (WVNS) may use direct additions to the melter to correct the problem. If the masses of the pertinent oxides in the melter are known, RECIPE has the ability to compute the amount of chemical additives required for this special feed. This computation is essentially identical to that of computing a shim for the CFMT or the MFT.

(a) West Valley Nuclear Services proposes to continue operation during the period that the sample is being analyzed. This method will be feasible as long as feed composition can be controlled reliably enough to keep the melter composition within acceptable limits. Its feasibility is beyond the scope of this report, as it does not affect the requirements for the RECIPE code.

GLASS AND FEED COMPOSITIONS

Tables 1 through 3 provide background information on glass composition. Table 1 gives the reference glass composition of WV-205 glass, obtained by a mass balance. West Valley's current strategy is to produce glass having a composition as close to the desired composition (reference glass) as feasible. Table 2 lists those elements from Table 1 that might be controlled and therefore are germane to the RECIPE code. These include the elements, exclusive of oxygen, that exist in the glass at concentrations greater than 0.5 wt% and hence must be monitored according to the waste acceptance specifications.[(a)] In addition, several elements that may affect the glass composition are included for completeness. Note that seven elements (B, K, Li, Mg, Na, Si and Ti) will be provided primarily (>80%) by glass formers; the waste will provide the bulk of the remainder (Al, Ca, Cr, Fe, Mn, Ni, P, S, Zr and "residual").

The percentages given in Table 2 can be considered to be the ideal or target composition of WV-205 glass (the reference glass at the time RECIPE was developed). However, the values of nominal weight percent oxide that are specified for the feed or the melter may not necessarily be the same as those given in Table 2. For example, if losses to the off-gas system are large for a certain oxide, the user may specify a higher value for that oxide than is given in Table 2.

It is anticipated that tolerances for each specified oxide concentration will be developed such that feeds with compositions within these limits will process acceptably and that the resulting glass will have adequate properties for waste qualification. A narrower range of tolerances almost certainly will be used in feed preparation and melter composition to allow for analytical and process measurement errors.

Table 3 presents the additives that will most likely be used for each of the oxides listed in Table 2. Other additives can be accommodated by RECIPE; the relevant inputs are described later.

(a) Office of Civilian Radioactive Waste Management. 1987. Waste Acceptance Preliminary Specifications for the West Valley Demonstration Project High-Level Waste Form. DOE/RW-0136, U.S. Department of Energy, Washington, D.C.

TABLE 1. Composition of WV-205 Glass (by mass balance)

Component	Weight Percent
AgO	0.0001
Al_2O_3	2.8295
AmO_2	0.0073
BaO	0.0540
B_2O_3	9.9516
CaO	0.5993
CdO	0.0003
CeO_2	0.0670
CmO_2	0.0001
CoO	0.0002
Cr_2O_3	0.3112
Cs_2O	0.0826
CuO	0.0001
Eu_2O_3	0.0014
Fe_2O_3	12.1573
Gd_2O_3	0.0003
In_2O_3	0.0001
K_2O	3.5733
La_2O_3	0.0337
Li_2O	3.0315
MgO	1.3032
MnO_2	1.3107
MoO_3	0.0088
NaCl	0.0183
NaF	0.0013
Na_2O	10.9335
Nd_2O_3	0.1209
NiO	0.3358
NpO_2	0.0224
P_2O_5	2.5084
PdO	0.0062

TABLE 1. contd

Component	Weight Percent
Pm_2O_3	0.0003
Pr_6O_{11}	0.0321
PuO_2	0.0076
Rb_2O	0.0005
RhO_2	0.0136
RuO_2	0.0759
SO_3	0.2164
Sb_2O_3	0.0001
SeO_2	0.0005
SiO_2	44.8770
Sm_2O_3	0.0267
SnO_2	0.0006
SrO	0.0269
Tc_2O_7	0.0021
ThO_2	3.5844
TeO_2	0.0028
TiO_2	0.9800
UO_2	0.5605
Y_2O_3	0.0177
ZnO	0.0010
ZrO_2	0.2943
Insolubles	0.0080
TOTAL	100.0000

TABLE 2. Elements Germane to the RECIPE Code

Element	I[a]	Oxide, Weight Percent
Al	1	2.83
B	2	9.95
Ca[b]	3	0.60
Cr[b]	4	0.31
Fe	5	12.16
K	6	3.57
Li	7	3.03
Mg	8	1.30
Mn	9	1.31
Na	10	10.94
Ni[b]	11	0.34
P	12	2.51
S[b]	13	0.22
Si	14	44.88
Th	15	3.58
Ti	16	0.98
U	17	0.56
Zr[b]	18	0.29
residual[c]	19	0.64
		100.00

(a) For illustration, each oxide has been assigned a unique value of I as shown here.

(b) These elements are found in concentrations of less than 0.5 wt% (of the element--not the oxide); therefore, they need not be monitored according to waste acceptance specifications.

(c) The entry "residual" in Table 2 is the sum of the concentrations of all oxides listed in Table 1 that are not listed in Table 2.

TABLE 3. Proposed Chemical Additives for Feed Preparation

Oxide	J[a]	Proposed Additives[b]
Al_2O_3	1	$Al(OH)_3$ or $Al(NO_3)_3*9H_2O$
B_2O_3	2	$Na_2B_4O_7*10H_2O$ (borax)[c] boric acid
CaO	3	To be identified
Cr_2O_3	4	To be identified
Fe_2O_3	5	$Fe(OH)_3$ or $Fe(NO_3)_39*H_2O$
K_2O	6	$K_2CO_3*1.5H_2O$ or KCOOH
Li_2O	7	Li_2CO_3 or LiCOOH
MgO	8	$Mg(OH)_2$
MnO_2	9	MnO(OH) (manganite) or $Mn(NO_3)_2*4H_2O$
Na_2O	10	NaCOOH or NaOH or $NaNO_3$
NiO	11	To be identified
P_2O_5	12	H_3PO_4 (water solution) or $FePO_42H_2O$
SO_3	13	To be identified
SiO_2	14	SiO_2 (finely ground silica sand)
ThO_2	15	$Zr(OH)_4$ or $ZrO(NO_3)_2*2H_2O$
TiO_2	16	TiO_2 (fine powder)
UO_2	17	$Zr(OH)_4$ or $ZrO(NO_3)_2*2H_2O$
ZrO_2	18	$Zr(OH)_4$ or $ZrO(NO_3)_2*2H_2O$
Residual	19	None

(a) For illustration, each source has been given a unique value of J.
(b) These additives may be industrial-grade or reagent-grade chemicals; it is assumed that WVNS will have knowledge of the impurities in terms of the oxides listed here.
(c) Borax contains significant amounts of sodium as well as boron, and would, therefore, be a partial source of sodium as well as boron.

3.0 MATHEMATICAL SOLUTION

The method of determining the required amounts of additives to bring the contents of a tank to a desired composition is described here. Before proceeding further, two terms should be defined: "additives" and "sources." The oxide-forming chemicals that will be added to the tank (such as those listed in Table 3), are referred to as "additives." The combination of additives supplying a particular oxide is referred to as a "source" of that particular oxide. The oxide Na_2O provides an example. The user will be required to supply RECIPE with the additives for Na_2O along with the weight percentages of the source that each additive will comprise (the relative percentage).[(a)] For example, if the user specifies that for every 100 kg of Na_2O source that is required, 10 kg of NACOOH, 50 kg of NaOH, and 40 kg of $NaNO_3$ are to be used (i.e., the relative percentages are 10%, 50%, and 40%), RECIPE will compute an "Na_2O-source" having a composition computed from the composition of the three "Na_2O-additives" and their relative percentages. Thus NaCOOH, NaOH and $NaNO_3$ are "Na_2O-additives" that make up the "Na_2O-source." The user can use other Na_2O-additives than those specified on Table 3; RECIPE only needs the chemical composition (in terms of weight percent of the oxides of the elements listed in Table 2) and the percentage of the Na_2O source that it will comprise. With this information we can proceed.

EQUATIONS (FOR WASTE IN THE CFMT)

We first consider the case of computing the additives necessary for the waste in the CFMT; then, we discuss the application of these equations to computing shims for either the MFT or the melter and, finally, the computation of corrective additions to the slurry in the CCMT or in a shim preparation tank. The following terms must be defined:

(a) The selection of the relative percentages of the additives that make up the sources will be made by the operator because RECIPE does not have algorithms for computing the requisite amounts of non-oxides such as formate.

imax	=	number of oxides to be controlled, dimensionless
jmax	=	number of sources (equal to imax), dimensionless
kmax	=	number of additives per source, dimensionless
RP(j,k)	=	relative percentage, the percent of the mass of the source j that is composed of additive k, dimensionless
S(j)	=	weight of source j to add to the CFMT to bring its composition to the desired value, kg
W(j,k)	=	weight of additive k from source j, kg
WP(i,j)	=	weight percent of oxide i in source j, dimensionless
WPA(i,j,k)	=	weight percent of oxide i in additive k for source j
WPO(i)	=	desired weight percent of oxide i in the glass, dimensionless[(a)]
WPT(i)	=	weight percent of oxide i in the CFMT after addition of the additives, dimensionless
WPTO(i)	=	weight percent of oxide i in the CFMT (i.e., in the waste) prior to addition of the sources (additives), dimensionless
WTM	=	final weight of contents of the CFMT (after sources have been added), kg
WTO	=	weight of contents of the CFMT when the sample used to determine WPTO(i) was taken, kg
X(j)	=	quantity defined by Equation (7).

We now consider the computation of the sources from a knowledge of the compositions of the additives. RECIPE has the capability of accepting a

---

(a) If there are no significant losses to the off-gas system, these concentrations will be the same as the glass. However, if losses to the off-gas system are significant, then the "glass composition," i.e., the values of WPO(i), may be adjusted accordingly to compensate.

maximum of 5 additives for each of the oxide sources.[a] Each of these additives will be assigned an index value of k. The user will input values of WPA(i,j,k) for all values of i, j and k. The composition of source i will be computed from the equation

$$WP(i,j) = \sum_{k=1}^{kmax} RP(j,k) * WPA(i,j,k)/100 \qquad (1)$$

Equation (1) reduces the problem of computing the required amount of additives to determining the amounts of each of the sources that, when added to the CFMT, will give a mixture whose composition is such that the ratios between the concentration of the oxides in the CFMT are the same as the ratios between the concentrations in the glass.

Because we wish to produce the least amount of glass per unit mass of waste, we will treat the contents of the CFMT as a source for the oxide (hereafter called the "pivot oxide") which has the highest concentration compared to that of the glass. By definition, the pivot oxide has the highest value of R'(i), where

$$R'(i) = WPTO(i)/WPO(i) \qquad (2)$$

The value of i (and j) for the maximum value of R' will be called i' and j' where

$$i' = j' \qquad (3)$$

The source for i', S(j'), is equal to WTO, the contents of the CFMT. Thus, no additives are required for the ith or pivot oxide.

Because the material in the CFMT is treated as a source of oxide i', the values of WP(i,j') must be set equal to WPTO(i). For example, if we

---

(a) RECIPE can handle up to 100 oxide sources; the 19 presented in Table 2 are the sources most likely to be actually controlled.

assume that we are dealing with the 19 sources given on Table 2 and we assume that R'(8) is the highest value of R'(i) (i.e., if MgO is the pivot oxide), then i'= j'= 8. As mentioned above, we will treat the contents of the CFMT as the source for MgO. This requires setting S(8) = WTO and setting the 19 row values in column 8 of [WP], i.e. all 19 of WP(i,8), equal to the CFMT oxide composition values in WPTO(i). Below, we examine the method used to compute the magnitude of the remaining sources.

The relationship between the composition of the sources, the amount of each source, the final weight of the contents of the CFMT (after the sources have been added), and the desired oxide concentrations is, for the ith oxide,

$$\sum_{j=1}^{j=jmax} WP(i,j) * S(j) = WPT(i) * WTM \qquad (4)$$

As mentioned above, RECIPE seeks to make the ratio of the concentration of the different oxides in the CFMT identical to the ratio between the concentrations of the oxides in the glass.[(a)] This is accomplished by incorporating the constraint on the ordered oxides:

$$WPT(i + 1)/WPT(i) = WPO(i + 1)/WPO(i) \qquad (5)$$

for i from 1 to imax (all oxides including the residual). This constraint is met when

$$WPT(i)/WPO(i) = K \qquad (6)$$

for all values of i.

We then define X(j), for the jth source, as

(a) If there are no losses to the off-gas system, these concentrations will be the same as the glass. However, if losses to the off-gas system are significant, then the "glass composition" used may be adjusted accordingly to compensate.

$$X(j) = S(j)/(K * WTM) \tag{7}$$

which, when inserted into equation (4), results in

$$\sum_{j=1}^{j=jmax} WP(i,j) * X(j) = WPO(i) \tag{8}$$

The set of imax equations (one for each i) can be written in matrix notation as

$$[WP] * [X] = [WPO] \tag{9}$$

The WP matrix and the WPO vector are fully defined, so we can solve for the X vector, thereby obtaining values of X(j). Values for the amounts of each source to add [i.e., S(j)] are given by Equation (12), whose derivation follows. From Equation (5) we can write

$$S(j)/S(j') = (X(j) * K * WT)/(X(j') * K * WT) = X(j)/X(j') \tag{10}$$

Rearranging gives

$$S(j) = S(j') * X(j)/X(j') \tag{11}$$

and since S(j') = WTO

$$S(j) = WTO * X(j)/X(j') \tag{12}$$

except for S(j') which, as mentioned above, is equal to WTO (no material outside of the tank need be added for the pivot oxide). The amounts of each additive are computed from the relation

$$W(j,k) = RP(j,k) * S(j)/100 \tag{13}$$

except that W(j',k) = 0 for all k because the additives for the pivot oxide i' already exist in the waste.

COMPUTATION OF SHIMS FOR THE MFT AND FOR THE MELTER

These computations are identical to the computation of the additives required for the CFMT. The only difference is that the mass of material and the weight percent of the controlled oxides in either the MFT or the melter are used instead of those of the CFMT.

COMPUTATION OF CORRECTIVE ADDITIONS TO THE SLURRY IN THE CCMT OR SHIM PREPARATION TANK

The computations described in the two previous sections give the amounts of the various sources to be added to the CCMT or to a shim preparation tank to prepare the slurry for either the CFMT, the MFT, or the melter. Once this slurry has been made up, its total mass will be measured (probably from volume and density), and it will be sampled and analyzed to determine its composition. RECIPE will then be used to determine whether the mass and composition of this slurry are acceptable and, if not, the amount and types of additives required to bring the contents and mass of the slurry to an acceptable level.

The following terms are defined at this point:

WPGO(i)	=	the weight percent of oxide i in the glass that would be formed if the slurry were mixed with the contents of the target tank and processed into glass, dimensionless
WPMO(i)	=	weight percent of oxide i in the mixture that would be obtained by combining the contents of the slurry preparation tank with the contents of the target tank, dimensionless
WPSO(i)	=	weight percent of oxide i in the slurry, dimensionless
WTM	=	weight of the mixture that would be obtained by combining the contents of the slurry preparation tank with the contents of the target tank, kg

WTS = weight of slurry, kg,

where "slurry" refers to the contents of the CCMT or the slurry preparation tank and "target tank" refers to the tank to which the slurry will be added (CFMT, MFT or melter).

The values of WPSO(i) will be available from the above mentioned chemical analysis. WPS is merely the sum of the weights of all of the additives given by

$$WTS = \sum_{j=1}^{j=jmax} \sum_{k=1}^{k=kmax} W(j,k) \tag{14}$$

Once the values of WPS and WTS are available, the following equations are used to compute WTM, WTMO(i) and WPG(i):

$$WTM = WTO + WTS \tag{15}$$

$$WPMO(i) = [WPTO(i) * WTO + WPSO(i) * WTS]/WTM \tag{16}$$

$$WPGO(i) = 100 * WPMO(i)/\sum_{i=1}^{imax} WPMO(i) \tag{17}$$

If all WPGO(i) are within tolerances, then the slurry is acceptable and can be added to the target tank. If not, RECIPE can be used to compute, by means of WTM, WPMO(i) and WPO(i), the amounts of additives that will have to be added to the CCMT to make the resulting mixture acceptable. With this information, the operator can decide whether it is best to add the material to the slurry preparation tank or whether it would be better to dump the tank and start over.

OTHER COMPUTATIONS

RECIPE can be used to compute the concentration of non-glass-forming oxides. These oxides can be any chemical species whose concentration is

not controlled, such as formates, carbonates, or nitrates. RECIPE expresses the concentration of these non-oxides in terms of the ratio of the non-oxide mass to both the glass and the feed mass.
We therefore define the following:

M(i")	=	the total mass of the i"th non-oxide in the target tank and in the slurry preparation tank, kg
MG	=	the mass of the glass that would be produced from the mass of feed created by mixing the material in the target tank and the slurry preparation tank, assuming that glass-forming oxides are lost to the off-gas system
RF(i")	=	the ratio of the mass of the i" non-oxide to the mass of the feed, dimensionless
RG(i")	=	the ratio of the mass of the i"th non-oxide to the mass of the glass, dimensionless.

The mass of the feed that would result from mixing the contents of the slurry preparation tank and the target tank is given by WTM (Equation 15). The mass of the ith oxide in the glass that would result from the feed produced by combining the contents of the two tanks is given by

$$M(i) = WPTO(i) * WTO/100 + \sum_{i=1}^{i=imax} \sum_{k=1}^{k=kmax} W(j,k) * WPA(i,j,k) \qquad (18)$$

Since the mass of glass produced is given by the sum of the masses of the glass-forming oxides, we obtain

$$MG = \sum_{i=1}^{i=imax} M(i) \qquad (19)$$

The mass of a given non-oxide found in the slurry preparation tank and the target tank is given by

$$M(i) = WPTO(i") * WTO/100 + \sum_{i"=1}^{i"=imax} \sum_{k=1}^{k=kmax} W(j,k) * WPA(i",j,k) \quad (20)$$

Thus RF(i") and RG(i") are given by

$$RF(i") = M(i)/WTM \quad (21)$$

and

$$RG(i") = M(i")/MG \quad (22)$$

The values of WPA(i",j,k) are included in the WPA array; the details of the process are shown in the description of the REC1PE input.

4.0 IMPLEMENTATION GUIDE AND INPUT DESCRIPTION

PART A: RECIPE REQUIREMENTS

The RECIPE software uses the LINPACK library, a set of FORTRAN routines for solving linear algebraic equations. Because LINPACK software is public domain, no licensing agreement is necessary for the use of the software. These routines are stored in two object libraries, LINPACK.OLB and BLAS.OLB, which must be linked with the main RECIPE program. For more information see the LINPACK user's guide.

RECIPE is implemented by means of object libraries. In addition to the LINPACK and BLAS libraries, we have created MODULE.OLB, which contains all RECIPE module object codes except for the main module (RECIPE.FOR). To use RECIPE, it must first be compiled using the following command:

```
$FORTRAN RECIPE
```

This command creates the object code (.OBJ). The next step is to link RECIPE with all the libraries. This is done by entering the following command:

```
$LINK RECIPE,MODULE/LIB,LINPACK/LIB,BLAS/LIB
```

This command creates the executable code (.EXE), and RECIPE is now ready to run. See Section 5.0 for information on running RECIPE.

PART B: RECIPE INPUT FILE FORMATS

The following discussion describes the format of the data files that are input to RECIPE. The files can be created by the RS/1 command procedures which will automatically put them in the format outlined below. The files may also be created by the VAX/VMS editor or other related software. If the files are not created with the RS/1 command procedures, they must be formatted according to the following specifications. The files contain certain areas where the program looks specifically for certain data. These

are called "data fields" in the description below. The description also refers to "blank" lines or spaces. Blank areas are skipped over by RECIPE; therefore, these areas may contain any notes or documentation that the user desires to store in the file. It is important that documentation is confined to these areas only. The format for the data files is quite rigid and must be strictly followed. There are no empty lines or spaces separating the data fields unless specifically described below. RECIPE has been designed to detect errors in the file formats and inconsistencies between the different files used. When RECIPE detects a problem, it provides an error message and stops. It is to the user's advantage to make sure that the files are of the correct format before RECIPE is used. An example of each file has been included after this discussion. The first line in each example gives the column number, and the first column in each line gives the row number.

General Notes

The name of each oxide, anion[(a)], and additive must be exactly the same regardless of which file it is used in or where it is used in the file.

Weight Percent of Oxides in Additives for Sources File (WPA)

This file contains a block for each oxide listed in the target composition. The block contains the name of the oxide, the names of the additives providing the oxide, and a listing of the oxide and anion contents of each additive.

- Lines 1 and 2 of the file are blank lines.
- Line 3 is the first line of block for the first oxide. It begins with an 8-character data field containing the name of the source. The rest of the line may then be considered a blank area.

(a) The term "anion" refers to the non-glass-forming oxides discussed in Chapter 3.

- Line 4 contains:
  - A 20-space blank area.
  - Five 8-character data fields containing the first half of the additive names for the source listed in Line 3. Each of the five data fields is followed by a 2-character blank field.
- Line 5 contains:
  - A 20-space blank area.
  - Five 8-character data fields containing the second half of the additive names for the source listed Line 3. Each of the five data fields is followed by a 2-character blank field.
- Line 6 is a blank line.
- Line 7 contains:
  - An 8-character data field containing an "O" or an "NO" depending on whether this line contains data for an oxide ("O") or an anion "NO". "O" and "NO" must be in upper-case characters and must be left justified. This field is followed by a 2-character blank field.
  - An 8-character data field containing the name of the oxide followed by a 2-character blank field.
  - Five 8-character data fields containing real numbers showing the amount of oxide making up the additive named in Lines 4 and 5. Each of the five data fields is followed by a 2-character blank field.
- Lines 8 and following are of the same format as Line 7. There should be one line for each oxide in the target composition, as well as one line for each anion the user wishes the code to track.
- When the data for one source have been supplied, the user begins the data for the next source, starting with the format of Line 1 and continuing. There must be one source block for each oxide in the target composition.

WPA INPUT FILE FORMAT:

```
          1         2         3         4         5         6         7         8
 12345678901234567890123456789012345678901234567890123456789012345678901234567890
 1
 2 ------------------------------------------------------------------------------
 3 Al2O3     Additives
 4                     Additive  Additive  Additive  Additive  Additive
 5 Type      Oxide     #1        #2        #3        #4        #5
 6 ====      =======   ========  ========  ========  ========  ======
 7 0         Al2O3       90.000    85.000    80.000    89.000    93.000
 8 0         B2O3         0.000     0.000     0.000     0.000     0.000
 9 0         CaO          0.000     0.300     0.000     0.000     0.500
10 0         Cr2O3        0.000     0.000     0.300     0.500     0.600
 1 0         FeO          0.000     0.000     0.000     0.000     0.000
 2 0         K2O          0.000     0.000     0.000     0.000     0.500
 3 0         Li2O         0.000     0.000     0.300     0.500     0.600
 4 0         MgO          0.000     0.500     0.000     0.500     0.400
 5 0         MnO          0.000     0.600     0.000     0.600     0.000
 6 0         Na2O         0.000     0.800     0.000     0.800     0.400
 7 0         NiO          0.000     0.500     0.000     0.500     0.020
 8 0         P2O5         0.000     0.400     0.000     0.400     0.330
 9 0         SO3          0.300     0.000     0.000     0.000     0.450
20 0         SiO2         0.000     0.400     0.000     0.400     0.670
 1 0         ThO2         0.000     0.020     0.300     0.800     0.670
 2 0         TiO2         0.000     0.330     0.000     0.500     0.000
 3 0         UO2          0.000     0.450     0.000     0.400     0.000
 4 0         ZrO2         0.000     0.00      0.000     0.00      0.500
 5 0         Residual     0.000     0.670     0.000     0.670     0.400
 6 NO        CO3          0.700     0.700     0.800     0.600     0.700
 7 NO        COOH         0.000     0.500     0.400     0.000     0.000
 8 NO        NO3          0.070     0.070     0.140     0.010     0.080
 9
30
```

1	B2O3	Additives					
2			Additive	Additive	Additive	Additive	Additive
3	Type	Oxide	#1	#2	#3	#4	#5
4	====	=======	==========	==========	==========	==========	==========
5	0	Al2O3	0.300	0.000	0.000	0.000	0.000
6	0	B2O3	93.000	55.000	70.000	65.000	80.000
7	0	CaO	0.000	0.300	0.000	0.000	0.500
8	0	Cr2O3	0.000	0.000	0.300	0.500	0.600
9	0	FeO	0.000	0.000	0.000	0.000	0.000
40	0	K2O	0.000	0.000	0.300	0.000	0.500
1	0	Li2O	0.000	0.000	0.300	0.500	0.600
2	0	MgO	0.000	0.500	0.000	0.500	0.400
3	0	MnO	0.000	0.600	0.000	0.600	0.000
4	0	Na2O	0.000	0.800	0.000	0.800	0.400
5	0	NiO	0.450	0.500	0.000	0.500	0.020
6	0	P2O5	0.000	0.400	0.000	0.400	0.330
7	0	SO3	0.300	0.000	0.000	0.000	0.450
8	0	SiO2	0.000	0.400	0.890	0.400	0.670
9	0	ThO2	0.000	0.020	0.300	0.800	0.670
50	0	TiO2	0.000	0.330	0.000	0.500	0.000
1	0	UO2	0.700	0.450	0.000	0.400	0.000
2	0	ZrO2	0.000	0.000	0.000	0.000	0.500
3	0	Residual	0.000	0.670	0.000	0.670	0.400
4	NO	CO3	0.870	0.700	0.800	0.600	0.700
5	NO	COOH	0.001	0.001	0.002	0.003	0.001
6	NO	NO3	0.070	0.070	0.140	0.010	0.080
7							
8							
9	CaO	Additives					
60			Additive	Additive	Additive	Additive	Additive
1	Type	Oxide	#1	#2	#3	#4	#5
2	====	=======	==========	==========	==========	==========	==========
3	0	Al2O3	0.000	0.000	0.300	0.400	0.500
4	0	B2O3	0.000	0.000	0.000	0.000	0.000
5	0	CaO	90.000	85.000	80.000	89.000	93.000
6	0	Cr2O3	0.000	0.000	0.300	0.500	0.600
7	0	FeO	0.000	0.000	0.000	0.000	0.000
8	0	K2O	0.000	0.000	0.000	0.000	0.500
9	0	Li2O	0.000	0.000	0.300	0.500	0.600
70	0	MgO	0.000	0.500	0.000	0.500	0.400
1	0	MnO	0.000	0.600	0.000	0.600	0.000
2	0	Na2O	0.000	0.800	0.000	0.800	0.400
3	0	NiO	0.870	0.500	0.900	0.500	0.020
4	0	P2O5	0.000	0.400	0.000	0.400	0.330
5	0	SO3	0.300	0.900	0.000	0.000	0.450
6	0	SiO2	0.000	0.400	0.000	0.400	0.670
7	0	ThO2	0.000	0.020	0.300	0.800	0.670

8	0	TiO2	0.000	0.330	0.000	0.500	0.000
9	0	UO2	0.000	0.450	0.000	0.400	0.000
80	0	ZrO2	0.000	0.000	0.000	0.000	0.500
1	0	Residual	0.000	0.670	0.000	0.670	0.400
2	NO	CO3	0.700	0.700	0.800	0.600	0.700
3	NO	COOH	0.000	0.010	0.000	0.155	0.000
4	NO	NO3	0.070	0.070	0.140	0.010	0.080

Additive Relative Percentages File (RP)

This file contains a block of data for each oxide source in the target composition. The block contains the name of the oxide, the names of the additives providing this oxide, and the percent of each additive to be used.

- Lines 1-6 are blank lines.
- Line 7 is the beginning of the block for the first oxide source. It is a blank line.
- Line 8 contains:
  - A 10-character blank field.
  - An 8-character data field containing the oxide name.
  - The rest of the line may be considered a blank area.
- Line 9 contains:
  - A 10-character blank field.
  - Five 8-character data fields containing the first half of the additive names. Each of the five data fields is followed by a 2-character blank field.
- Line 10 contains:
  - A 10-character blank field.
  - Five 8-character data fields containing the second half of the additive names. Each of the five data fields is followed by a 2-character blank field.
- Line 11 contains:
  - A 10-character blank field.

- Five 8-character data fields containing real numbers showing the percent of each additive in the source listed in Line 8. Each of the five data fields is followed by a 2-character blank field.

- The information for the each source is then supplied, following the formats specified in Lines 7 through 11. There must be a block of data for each oxide in the target composition.

RP INPUT FILE FORMAT:

```
          1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
 1
 2 RP  100R x 5C                                          20-OCT-86 18:02 Page 1
 3
 4                      Data from file rp
 5
 6 0      1           2         3         4         5
 7   6
 8   7        Al2O3
 9   8        Additive  Additive  Additive  Additive  Additive
10   9        #1        #2        #3        #4        #5
 1  10              5.        10.       15.       30.       40.
 2  11
 3  12        B2O3
 4  13        Additive  Additive  Additive  Additive  Additive
 5  14        #1        #2        #3        #4        #5
 6  15              5.        10.       15.       30.       40.
 7  16
 8  17        CaO
 9  18        Additive  Additive  Additive  Additive  Additive
20  19        #1        #2        #3        #4        #5
 1  20              5.        10.       15.       30.       40.
 2  21
 3  22        Cr2O3
 4  23        Additive  Additive  Additive  Additive  Additive
 5  24        #1        #2        #3        #4        #5
 6  25              5.        10.       15.       30.       40.
 7  26
 8  27        FeO
 9  28        Additive  Additive  Additive  Additive  Additive
30  29        #1        #2        #3        #4        #5
 1  30              5.        10.       15.       30.       40.
 2  31
```

```
 3  32        K20
 4  33        Additive  Additive  Additive  Additive  Additive
 5  34        #1        #2        #3        #4        #5
 6  35              5.       10.       15.       30.       40.
 7  36
 8  37        Li20
 9  38        Additive  Additive  Additive  Additive  Additive
40  39        #1        #2        #3        #4        #5
 1  40             20.       20.       10.       40.       10.
 2  41
 3  42        MgO
 4  43        Additive  Additive  Additive  Additive  Additive
 5  44        #1        #2        #3        #4        #5
 6  45             18.       12.       20.        5.       45.
 7  46
 8  47        MnO
 9  48        Additive  Additive  Additive  Additive  Additive
50  49        #1        #2        #3        #4        #5
 1  50             25.       20.       30.        0.       25.
 2  51
 3  52        Na20
 4  53        Additive  Additive  Additive  Additive  Additive
 5  54        #1        #2        #3        #4        #5
 6  55              5.       25.        5.       35.       30.
 7  56
 8  57        NiO
 9  58        Additive  Additive  Additive  Additive  Additive
60  59        #1        #2        #3        #4        #5
 1  60             10.       20.       30.       40.        0.
 2  61
 3  62        P205
 4  63        Additive  Additive  Additive  Additive  Additive
 5  64        #1        #2        #3        #4        #5
 6  65              0.       40.       30.       20.       10.
 7  66
 8  67        S03
 9  68        Additive  Additive  Additive  Additive  Additive
70  69        #1        #2        #3        #4        #5
 1  70            100.        0.        0.        0.        0.
 2  71
 3  72        Si02
 4  73        Additive  Additive  Additive  Additive  Additive
 5  74        #1        #2        #3        #4        #5
 6  75              2.        8.       10.       30.       50.
 7  76
 8  77        Th02
 9  78        Additive  Additive  Additive  Additive  Additive
80  79        #1        #2        #3        #4        #5
 1  80             35.       25.       10.       25.        5.
```

```
 2  81
 3  82       TiO2
 4  83       Additive  Additive  Additive  Additive  Additive
 5  84       #1        #2        #3        #4        #5
 6  85            5.       25.       10.       25.       35.
 7  86
 8  87       UO2
 9  88       Additive  Additive  Additive  Additive  Additive
90  89       #1        #2        #3        #4        #5
 1  90            18.       12.       15.       15.       40.
 2  91
 3  92       ZrO2
 4  93       Additive  Additive  Additive  Additive  Additive
 5  94       #1        #2        #3        #4        #5
 6  95            8.       22.        5.       30.       35.
 7  96
 8  97       Residual
 9  98       Additive  Additive  Additive  Additive  Additive
91  99       #1        #2        #3        #4        #5
 1 100            1.        9.       20.       40.       30.
```

Tank Composition File

This file contains the weight percent composition of the oxides and anions in the tank.

- Lines 1-7 are blank lines.
- Line 8 contains:
  - A 10-character blank field.
  - An 8-character data field containing the oxide name followed by a 2-character blank field.
  - An 8-character data field containing weight percent of this oxide in the tank composition specified as a real number.
- Lines 9 and following are the same format as Line 8. There is one line for each oxide in the target composition and each anion the user wishes to track through the program.

WPTO INPUT FILE FORMAT:

```
          1         2         3         4         5         6         7         8
  12345678901234567890123456789012345678901234567890123456789012345678901234567890
 1
 2 WPTO  28R x 3C                                          20-OCT-86 18:16 Page 1
 3
 4            Data from file wpto
 5
 6 0     1           2           3
 7  6    ==========  ==========
 8  7        Al203      .1587000
 9  8        B203       .3480198
10  9        CaO        .0286969
 1 10        Cr203      .0044047
 2 11        FeO        .5501324
 3 12        K20        .2198765
 4 13        Li20       .2330998
 5 14        MgO        .0547839
 6 15        MnO        .0686938
 7 16        Na20       .6298574
 8 17        NiO        .0073829
 9 18        P205       .1643920
20 19        S03        .0003892
 1 20        Si02       2.244392
 2 21        Th02       0.193850
 3 22        Ti02       0.093820
 4 23        U02        0.039285
 5 24        Zr02       0.000000
 6 25        Residual   0.012484
 7 26        C03        1.1
 8 27        COOH       1.2
 9 28        N03        1.4
```

Target Composition File

This file contains the amount of oxides expressed in weight percents in the target composition. It also contains lower and upper bounds specified in weight percents for each oxide.

- Lines 1-7 are blank lines.
- Line 8 contains:
  - A 10-character blank field.
  - An 8-character data field containing the oxide name followed by a 2-character blank field.

- An 8-character data field containing the minimum allowable weight percent for this oxide in the target composition specified as a real number. It is followed by a 2-character blank field.
- An 8-character data field containing the weight percent for this oxide in the target composition expressed as a real number. It is followed by a 2-character blank field.
- An 8-character data field containing the maximum allowable weight percent for this oxide in the target composition specified as a real number.

• Lines 9 and following are the same format as Line 8. There is one line for each oxide in the target composition.

WPO INPUT FILE FORMAT:

```
          1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
1                   Data from file wpo
2 0     1           2           3           4
3 ------------------------------------------------------
4  1    TARGET      GLASS       COMPOSITION
5  2
6  3
7  6    ==========  ==========  ==========  ==========
8  7        Al2O3     2.0         2.83        4.0
9  8        B2O3      9.0         9.95        12.0
10 9        CaO       0.0         0.60        2.0
1  10       Cr2O3     0.0         0.31        0.5
2  11       FeO       10.0        12.16       14.0
3  12       K2O       2.5         3.57        4.5
4  13       Li2O      2.0         3.03        3.5
5  14       MgO       0.5         1.30        2.0
6  15       MnO       0.0         1.31        2.0
7  16       Na2O      9.0         10.94       13.0
8  17       NiO       0.0         0.34        1.0
9  18       P2O5      0.0         2.51        3.0
20 19       SO3       0.0         0.22        0.5
1  20       SiO2      41.0        44.88       48.0
2  21       ThO2      0.0         3.58        4.5
3  22       TiO2      0.0         0.98        4.0
4  23       UO2       0.0         0.56        3.0
5  24       ZrO2      0.0         0.29        3.0
6  25       Residual  0.0         0.64        3.0
```

Slurry Composition File

The slurry composition file contains the weight percent of each oxide and anion in the slurry.

- Lines 1-6 are blank lines.
- Line 7 contains:
  - A 10-character blank field.
  - An 8-character field containing an "O" or an "NO" specifying whether this record is for an oxide ("O") or an anion ("NO"). This is followed by a 2-character blank field.
  - An 8-character field containing the oxide or anion name. This is followed by a 2-character blank field.
  - An 8-character field containing the weight percent of the oxide or anion in the slurry composition. This value should be expressed as a real number.
- Lines 8 and following have identical formats. There is one line for each oxide in the target composition, and for each anion the user wishes to track through the program.

WPSO INPUT FILE.DAT

```
          1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
 1
 2
 3
 4
 5
 6      ==========  ==========
 7           0         Al203     0.641567
 8           0         B203      2.787433
 9           0         CaO       0.169375
10           0         Cr203     0.022967
 1           0         FeO       3.644672
 2           0         K20       0.680114
 3           0         Li20      0.261507
 4           0         MgO       0.417848
 5           0         MnO       0.329250
```

6	0	Na2O	2.369485
7	0	NiO	0.155286
8	0	P2O5	0.406961
9	0	SO3	0.058978
20	0	SiO2	6.045949
1	0	ThO2	1.454600
2	0	TiO2	0.302297
3	0	UO2	0.072490
4	0	ZrO2	0.033074
5	0	Residual	0.302297
6	NO	CO3	1.100000
7	NO	COOH	1.200000
8	NO	NO3	1.400000

5.0 OPERATION OF RECIPE

This part of the user guide deals with actually running RECIPE interactively. The section contains two examples, which have been designed to cover all available options and results of the program as well as program responses to bad input or operator error. The user will benefit most by using this section as a tutorial while actually running the RECIPE code.

EXAMPLE 1

RECIPE requires the user to respond to directives and/or questions while using the program. Some of the user responses will require data file names. The following file names will be used for example 1 and appear in order of use:

WPATEST1.DAT - file containing additives identification and composition;

RPTEST1.DAT - file containing relative percentages of additives;

WPTOTEST1.DAT - file containing composition of tank contents;

WPOTEST1.DAT - file containing desired weight percent of oxides.

The weight of the tank contents (10000 Kg) and the specific gravity of the desired glass are also required (1.0).

Part A: Selecting an Option

After RECIPE has been properly compiled and linked (see Section 4.0), the command to execute the program should be typed and entered as follows:

$ RUN RECIPE<CR>

This will cause the RECIPE program to begin executing. The menu will appear, displaying the options that the user may select.

```
*******************************************************
*                                                     *
*   Welcome to RECIPE Version 1.0                     *
*                                                     *
*   Would you like to:                                *
*    1) Compose a new slurry or shim                  *
*    2) Check existing slurry for corrections         *
*    3) Terminate the program                         *
*                                                     *
*   Please enter the number of your selection.        *
*                                                     *
*******************************************************
```

This example illustrates option 1. It will require entering the number 1 in response to the directive, "Please enter the number of your selection." However, suppose that the number 4 is acidentally typed and entered before the mistake is noticed.

```
4<CR>
```

The following message is displayed:

```
4 is not an option.
Please try again.
```

and the menu is redisplayed.

```
*******************************************************
*                                                     *
*   Welcome to RECIPE Version 1.0                     *
*                                                     *
*   Would you like to:                                *
*    1) Compose a new slurry or shim                  *
*    2) Check existing slurry for corrections         *
*    3) Terminate the program                         *
*                                                     *
*   Please enter the number of your selection.        *
*                                                     *
*******************************************************
```

A very similar message will appear if any alphabetic character or symbol is entered.

```
a<CR>

Your input is not acceptable - try again.

*******************************************************
*                                                     *
*   Welcome to RECIPE Version 1.0                     *
*                                                     *
*   Would you like to:                                *
*    1) Compose a new slurry or shim                  *
*    2) Check existing slurry for corrections         *
*    3) Terminate the program                         *
*                                                     *
*   Please enter the number of your selection.        *
*                                                     *
*******************************************************
```

RECIPE will not allow any illegitimate choices to be entered as the result of a directive it has given to the user or inquiry it has made. When RECIPE detects bad input, it will continue to reprompt the user until a legitimate option is selected. This fact will not be repeated again in this section; however, the user should feel free to experiment with this aspect of the program.

Part B: Entering Weight Percents for the Additives

Once the number 1 has sucessfully been entered, the following directive is encountered:

```
1<CR>

Enter file containing additives identification and composition:
```

The program will now wait for the name of the file described to be entered. Once again, the correct file name for this example is WPATEST1.DAT. Suppose a typing error were made and the name of the file was entered as WPATERT1.DAT.

WPATERT1.DAT<CR>

If the incorrect file name does NOT exist in the current working directory, the following message will appear:

>>> ERROR Opening WPATERT1.DAT

and the file name directive will be redisplayed.

Enter file containing additives identification and composition:

If, however, the incorrect name typed is the name of some other file that exists in the directory,

WPATERT1.DAT<CR>

the file name is repeated, and the following inquiry is made:

Additives File: WPATERT1.DAT

Is this the correct file? Y or N:

If the mistake is not noticed and the response Y or y is entered,

Y<CR>

the program will the check the format of the file to make sure the correct file is being read. Since in this case the file name is incorrect and does not have the proper format, the following message is displayed:

>>> ERROR reading WPATERT1.DAT

and the execution of the program is halted.

```
FORTRAN STOP
$
```

The user is now back at the operating system level (VMS). At this point, he/she should check the input file and restart the program.

If, at the time, the incorrect file name is typed,

Enter file containing additives identification and composition:

WPATERT1.DAT<CR>

and the inquiry is received,

Additives File: WPATERT1.DAT

Is this the correct file? Y or N:

the user notices the typing error, and responds with a N or n,

N<CR>

the program will redisplay the file name directive.

Enter file containing additives identification and composition:

This time the correct file name is typed and the query is answered affirmatively. Since this is the correct file, the program will then read the contents of the file.

WPATEST1.DAT<CR>

Additives File: WPATEST1.DAT

Is this the correct file? Y or N:

Y<CR>

Part C: Entering Relative Percentages of the Additives

Once this step has been completed, the directive requesting the file containing the relative percentages of the additives is displayed,

Enter file containing relative percentages of additives:

and an identical process is performed. The correct file name is RPTEST1.DAT.

RPTEST1.DAT<CR>

Relative Percentages File: RPTEST1.DAT

Is this the correct file? Y or N:

Y<CR>

Part D: Entering the Tank Composition

After the contents of the relative percentages file are read, the program asks for the file containing the weight percent of oxides in the tank.

Enter file containing composition of tank contents:

The correct file name for this example is WPTOTEST1.DAT. Once again, if the file name is typed incorrectly

WPTOTERT1.DAT<CR>

and does not exist in the directory, the following message is displayed:

>>> ERROR Opening WPTOTERT1.DAT

```
Enter file containing composition of tank contents:
```

If the incorrect file name is actually a name of a file in the directory,

```
WPTOTERT1.DAT<CR>
```

the program will attempt to read the contents. However, if the file does not have the expected format, the following will be displayed:

```
>>> ERROR Detected in WPTOTERT1.DAT
>>> Incorrect File Format
```

and the program halts execution. Once again, the user will have to restart the program.

```
FORTRAN STOP
$
```

When the correct file name is entered,

```
WPTOTEST1.DAT<CR>
```

the file contents are displayed,

```
Tank File:  WPTOTEST1.DAT


Table of Tank Composition
Units are Weight Percent

Oxide/Anion          Tank
   Names         Composition

Al203               0.1587
B203                0.3480
CaO                 0.0287
Cr203               0.0044
FeO                 0.5501
K20                 0.2199
Li20                0.2331
MgO                 0.0548
```

MnO	0.0687
Na2O	0.6299
NiO	0.0074
P2O5	0.1644
SO3	0.0004
SiO2	2.2444
ThO2	0.1938
TiO2	0.0938
UO2	0.0393
ZrO2	0.0000
Residual	0.0125
CO3	1.1000
COOH	1.2000
NO3	1.4000

and the inquiry

Is this the correct file? Y or N:

is given. If a response of N or n is given,

N<CR>

the user is reprompted for the file name.

Enter file containing composition of tank contents:

This time, since the file name is correct, if a Y or y is typed in response to the inquiry,

WPTOTEST1.DAT<CR>

Tank File: WPTOTEST1.DAT

```
Table of Tank Composition
Units are Weight Percent

Oxide/Anion        Tank
   Names       Composition

Al2O3             0.1587
B2O3              0.3480
CaO               0.0287
Cr2O3             0.0044
FeO               0.5501
K2O               0.2199
Li2O              0.2331
MgO               0.0548
MnO               0.0687
Na2O              0.6299
NiO               0.0074
P2O5              0.1644
SO3               0.0004
SiO2              2.2444
ThO2              0.1938
TiO2              0.0938
UO2               0.0393
ZrO2              0.0000
Residual          0.0125
CO3               1.1000
COOH              1.2000
NO3               1.4000

Is this the correct file? Y or N:

Y<CR>
```

the program will continue. It now requests the weight of the mixture in the tank.

```
Enter weight of tank contents (in kilograms):
```

The number of kilograms is entered

```
10000<CR>
```

and the user is asked to double-check the value.

Weight of tank contents is 10000.0000 kilograms. Is this correct? Y or N:

Because this is the correct weight, an affirmative response is entered.

Y<CR>

Part E: Entering the Desired Oxide Weight Percentages

The last file that RECIPE needs as input is the file containing the desired weight percent of oxides in the resulting glass. The correct name of this file is WPOTEST1.DAT.

Enter file containing desired glass composition and tolerances:

WPOTEST1.DAT<CR>

Once again, the program displays the contents of the file whose name was just entered and asks the user to verify whether or not this is the correct file.

Glass Composition File: WPOTEST1.DAT

Desired Glass Tolerances and Composition Table
Units are Weight Percent

Oxide	Minimum Acceptable	Target	Maximum Acceptable
Al203	2.0000	2.8300	4.0000
B203	9.0000	9.9500	12.0000
CaO	0.0000	0.6000	2.0000
Cr203	0.0000	0.3100	0.5000
FeO	10.0000	12.1600	14.0000
K20	2.5000	3.5700	4.5000
Li20	2.0000	3.0300	3.5000
MgO	0.5000	1.3000	2.0000
MnO	0.0000	1.3100	2.0000
Na20	9.0000	10.9400	13.0000
NiO	0.0000	0.3400	1.0000
P205	0.0000	2.5100	3.0000

SO3	0.0000	0.2200	0.5000
SiO2	41.0000	44.8800	48.0000
ThO2	0.0000	3.5800	4.5000
TiO2	0.0000	0.9800	4.0000
UO2	0.0000	0.5600	3.0000
ZrO2	0.0000	0.2900	3.0000
Residual	0.0000	0.6400	3.0000

Is this the correct file? Y or N:

This is indeed the correct file, so a Y or y is entered.

Y<CR>

At this point, RECIPE needs to know the specific gravity of the desired glass.

Enter the specific gravity of the desired glass:

The correct response 1.0 is entered,

1.0<CR>

and the user is asked to verify his/her response.

The specific gravity of the desired glass is 1.0000.
Is this correct? Y or N:

This value is correct, so the user answers affirmatively.

Y<CR>

Part F: Continuing with RECIPE

All the necessary inputs that RECIPE needs to perform its job have now been entered. From this point until option 1 is completed, RECIPE will calculate and display for the user the data he/she needs in order to compose a new slurry.

The data are displayed in tabular form with appropriate headings that describe the table's contents. Once a table is displayed, it is followed by a directive to the user to press the RETURN key when he/she has read the information and would like to continue the program. The program will not continue until the RETURN key is pressed.

In addition to the visual display, an output file is created that contains all user inputs and RECIPE calculations. These files are named OUTPUT_DD_MMM_YY_HH_MM.DAT, where DD_MMM_YY is the day, month, year and HH_MMM is the hour and minute (taken from the system clock) that the run was made.

The following data represent an example of the visual display seen under option 1 of the RECIPE code. For information on the output file created, see TEST CASE 1 under Section 6.

Tank Composition to Desired Glass Oxide Ratios (R's):

Al203	0.0561	Na20	0.0576
B203	0.0350	NiO	0.0217
CaO	0.0478	P205	0.0655
Cr203	0.0142	SO3	0.0018
FeO	0.0452	SiO2	0.0500
K20	0.0616	ThO2	0.0541
Li20	0.0769	TiO2	0.0957
MgO	0.0421	UO2	0.0702
MnO	0.0524	ZrO2	0.0000
		Residual	0.0195

Pivot Oxide: TiO2

The pivot oxide is that oxide having the highest concentration in the tank compared to the concentration in the desired glass (highest R'). Therefore, no additives for that oxide are provided.

Press RETURN to continue

<CR>

Table of exact solutions

Index	Oxide	Exact solution - X(J)
1	Al2O3	0.1111E-01
2	B2O3	0.9245E-01
3	CaO	0.1846E-02
4	Cr2O3	0.9002E-03
5	FeO	0.1033E+00
6	K2O	0.2511E-01
7	Li2O	0.1151E-01
8	MgO	0.8945E-02
9	MnO	0.7705E-02
10	Na2O	0.7729E-01
11	NiO	0.1913E-02
12	P2O5	0.1337E-01
13	SO3	0.2721E-02
14	SiO2	0.3533E+00
15	ThO2	0.2195E-01
16	TiO2	0.8968E+01
17	UO2	0.1212E-02
18	ZrO2	0.1513E-02
19	Residual	0.2773E-02

Press RETURN to continue

<CR>

Index	Oxide	Amount of source - Kilograms
1	Al2O3	12.3896
2	B2O3	103.0822
3	CaO	2.0586
4	Cr2O3	1.0038
5	FeO	115.1942
6	K2O	28.0009
7	Li2O	12.8382
8	MgO	9.9745
9	MnO	8.5913
10	Na2O	86.1876
11	NiO	2.1330
12	P2O5	14.9100
13	SO3	3.0335
14	SiO2	393.9817
15	ThO2	24.4788
16	TiO2	0.0000
17	UO2	1.3516
18	ZrO2	1.6872
19	Residual	3.0924

Press RETURN to continue

<CR>

Table of Additives

Amounts of each additive, in kilograms, needed to create the desired glass composition.

Source	Additive #1	Additive #2	Additive #3	Additive #4	Additive #5
Al203	0.6195	1.2390	1.8584	3.7169	4.9558
B203	5.1541	10.3082	15.4623	30.9247	41.2329
CaO	0.1029	0.2059	0.3088	0.6176	0.8235
Cr203	0.0502	0.1004	0.1506	0.3011	0.4015
FeO	5.7597	11.5194	17.2791	34.5583	46.0777
K20	1.4000	2.8001	4.2001	8.4003	11.2004

Source	Additive #1	Additive #2	Additive #3	Additive #4	Additive #5
Li2O	2.5676	2.5676	1.2838	5.1353	1.2838

Source	Additive #1	Additive #2	Additive #3	Additive #4	Additive #5
MgO	1.7954	1.1969	1.9949	0.4987	4.4885

Source	Additive #1	Additive #2	Additive #3	Additive #4	Additive #5
MnO	2.1478	1.7183	2.5774	0.0000	2.1478

Source	Additive #1	Additive #2	Additive #3	Additive #4	Additive #5
Na2O	4.3094	21.5469	4.3094	30.1656	25.8563

Source	Additive #1	Additive #2	Additive #3	Additive #4	Additive #5
NiO	0.2133	0.4266	0.6399	0.8532	0.0000

Source	Additive #1	Additive #2	Additive #3	Additive #4	Additive #5
P2O5	0.0000	5.9640	4.4730	2.9820	1.4910

Source	Additive #1	Additive #2	Additive #3	Additive #4	Additive #5
SO3	3.0335	0.0000	0.0000	0.0000	0.0000

Source	Additive #1	Additive #2	Additive #3	Additive #4	Additive #5
SiO2	7.8796	31.5185	39.3982	118.1945	196.9908

Source	Additive #1	Additive #2	Additive #3	Additive #4	Additive #5
ThO2	8.5676	6.1197	2.4479	6.1197	1.2239

Source	Additive #1	Additive #2	Additive #3	Additive #4	Additive #5
TiO2	0.0000	0.0000	0.0000	0.0000	0.0000

Source	Additive #1	Additive #2	Additive #3	Additive #4	Additive #5
UO2	0.2433	0.1622	0.2027	0.2027	0.5406

Source	Additive #1	Additive #2	Additive #3	Additive #4	Additive #5
ZrO2	0.1350	0.3712	0.0844	0.5062	0.5905

Source	Additive #1	Additive #2	Additive #3	Additive #4	Additive #5
Residual	0.0309	0.2783	0.6185	1.2369	0.9277

Press RETURN to continue

<CR>

Table of Glass Tolerances and Calculated Glass Composition
(Units in Weight Percent)

Note: If any of the X(j)'s have been set to zero, the resulting glass will deviate from the desired glass composition.

Oxide	Minimum Tolerance	*WPG or WPGO	Desired Glass	Maximum Tolerance	In/Out Bounds
Al203	2.0000	2.8300	2.8300	4.0000	OK
B203	9.0000	9.9500	9.9500	12.0000	OK
CaO	0.0000	0.6000	0.6000	2.0000	OK
Cr203	0.0000	0.3100	0.3100	0.5000	OK
FeO	10.0000	12.1600	12.1600	14.0000	OK
K20	2.5000	3.5700	3.5700	4.5000	OK
Li20	2.0000	3.0300	3.0300	3.5000	OK
MgO	0.5000	1.3000	1.3000	2.0000	OK
MnO	0.0000	1.3100	1.3100	2.0000	OK
Na20	9.0000	10.9400	10.9400	13.0000	OK
NiO	0.0000	0.3400	0.3400	1.0000	OK
P205	0.0000	2.5100	2.5100	3.0000	OK

SO3	0.0000	0.2200	0.2200	0.5000	OK
SiO2	41.0000	44.8800	44.8800	48.0000	OK
ThO2	0.0000	3.5800	3.5800	4.5000	OK
TiO2	0.0000	0.9800	0.9800	4.0000	OK
UO2	0.0000	0.5600	0.5600	3.0000	OK
ZrO2	0.0000	0.2900	0.2900	3.0000	OK
Residual	0.0000	0.6400	0.6400	3.0000	OK

*Computed glass composition after mixing additives with the tank and assuming no losses from off gas.

Press RETURN to continue

<CR>

Anion	Ratio of anion mass to mass of glass
CO3	0.1037
COOH	0.1076
NO3	0.1261

Anion	Ratio of anion mass to mass of tank contents after addition of all additives to the tank.
CO3	0.0107
COOH	0.0111
NO3	0.0130

Volume of glass that would be produced: 1115.06

Press RETURN to continue

<CR>

At this point option 1 is completed. The main menu is redisplayed, and the number 3 will be entered to end this example.

```
*******************************************************
*                                                     *
*   Welcome to RECIPE Version 1.0                     *
*                                                     *
*   Would you like to:                                *
*    1) Compose a new slurry or shim                  *
*    2) Check existing slurry for corrections         *
*    3) Terminate the program                         *
*                                                     *
*   Please enter the number of your selection.        *
*                                                     *
*******************************************************

     3<CR>
     $
```

Part G: Occurrence of Negative Exact Solution Values

If negative values occur in the exact solution, RECIPE gives the user the option of changing some or all of the negative values to zero. The user input section of the code remains exactly the same as explained in PART A through PART E of this example. The following examples illustrate the parts of the code that differ from the remainder of the code (PART F). The most obvious difference is that negative values will be apparent in the table displaying the exact solution. This example contains two such values.

Table of exact solutions

Index	Oxide	Exact solution - X(J)
1	Al203	0.2297E-01
2	B203	0.1338E+00
3	CaO	0.3073E-02
4	Cr203	-0.8116E-03
5	FeO	0.1736E+00
6	K20	0.5720E-01
7	Li20	0.4533E-01
8	MgO	0.1323E-01
9	MnO	0.1458E-01
10	Na20	0.1608E+00
11	NiO	0.6195E-03
12	P205	0.3447E-01
13	SO3	-0.2040E-02

14	SiO2	0.6398E+00
15	ThO2	0.4336E-01
16	TiO2	0.1204E-01
17	UO2	0.4939E-02
18	ZrO2	0.2467E-01
19	Residual	0.5391E-03

Note: A negative value for X(j) is equivalent to the requirement that additives for the source be removed from the tank.

Press RETURN to continue

<CR>

Now the code requires the user to decide between zeroing all negative values or choosing specific negative values to zero.

You have a choice of options:

(1) Zero all negative values of [X]

(2) Select which values of [X] to zero

Please enter the number of your choice.

Part G.1: Selecting Which Values to Zero

In this example, option 2 is demonstrated, in which the user selects particular negative values to zero. The user enters the number 2.

2<CR>

The program then displays a table of all negative values.

Table of negative exact solution values

Index	Oxide	Exact solution - X(J)
4	Cr2O3	-0.8116E-03
13	SO3	-0.2040E-02

The user is then asked for the total number of negative oxide values being selected.

Please enter the number of total selections you will be making.

As one might expect, RECIPE will not allow any total to be entered that would be greater than the number of negative values it has displayed for the user. In this example, the user attempts to enter a total of three negative values, which is not acceptable because only two values are displayed as being negative.

3<CR>

The program repeats the unacceptable total along with an appropriate message.

3 is not an acceptable choice.
Please try again.

The table listing the appropriate selections is redisplayed, and once again the user is asked to enter a selection total.

Table of negative exact solution values

Index	Oxide	Exact solution - X(J)
4	Cr203	-0.8116E-03
13	S03	-0.2040E-02

Please enter the number of total selections you will be making.

Suppose the user wishes to zero one negative value but accidentally misskeys and enters the number 2.

2<CR>

The program will automatically double check the user's total for just such a typing error.

```
You have chosen to make    2 selections -
Is this correct? Type y or n:
```

In this case the entry was indeed an error, so the user responds negatively. Once again, the table is displayed, and the user is asked for the selection total. At this point, the user will correctly enter the number 1 and respond affirmatively to the double check.

```
N<CR>

    Table of negative exact solution values

Index        Oxide                    Exact solution - X(J)
-----        -----                    ---------------------
  4          Cr203                          -0.8116E-03
 13          S03                            -0.2040E-02

Please enter the number of total selections you will be making.

1<CR>

You have chosen to make    1 selections -
Is this correct? Type y or n:
Y<CR>
```

At this point, the negative values are displayed once again. The table contains an unique index, which is associated with each oxide. This is how RECIPE identifies which oxide(s) the user wisher to zero.

```
    Table of negative exact solution values

Index        Oxide                    Exact solution - X(J)
-----        -----                    ---------------------
  4          Cr203                          -0.8116E-03
 13          S03                            -0.2040E-02
```

And indeed, the next step is for the user to enter the index number of the oxides he/she wishes to zero. In this example, SO3 will be zeroed (index #13). The indexes must be entered exactly as stated - separated by commas. Once again RECIPE will only accept indexes that have been displayed. Here the user demonstrates what will occur if an invalid choice is made. (It should also be noted that RECIPE will only accept the number of indexes that was indicated by the user's previous total of selections.)

Now please enter the index number of each negative oxide you have chosen. The index values must be separated by a comma. For example: 1,4,19

20<CR>

The following selections you made are invalid:
20
Please enter all selections again.

Table of negative exact solution values

Index	Oxide	Exact solution - X(J)
4	Cr203	-0.8116E-03
13	SO3	-0.2040E-02

Now please enter the index number of each negative oxide you have chosen. The index values must be separated by a comma. For example: 1,4,19

After a valid choice is selected (either 4, 13 or 4,13) the user is asked to double check the entry. In this instance, the response is affirmative.

13<CR>

```
You have chosen the following selections -
13
Is this correct? Type y or n:
Y<CR>
```

From this point on, RECIPE works as usual.

Index	Oxide	Amount of source - Kilograms
1	Al2O3	9310.4736
2	B2O3	54237.2070
3	CaO	1245.7639
4	Cr2O3	-328.9919
5	FeO	70389.8125
6	K2O	23188.3066
7	Li2O	18373.4180
8	MgO	5362.8267
9	MnO	5908.4297
10	Na2O	65175.6172
11	NiO	251.1101
12	P2O5	13971.8291
13	SO3	0.0000
14	SiO2	259360.8594
15	ThO2	17575.0117
16	TiO2	4878.9946
17	UO2	2001.9706
18	ZrO2	0.0000
19	Residual	218.5536

```
Press RETURN to continue

<CR>
```

Part G.2: Zeroing All Negative Values

If the choice of options for negative values is displayed, the user may wish to zero all negative values. To do this, he/she enters the number 1 to the directive that is given. RECIPE then zeros all negative exact solution. This is transparent to the user.

You have a choice of options:

(1) Zero all negative values of [X]

(2) Select which values of [X] to zero

Please enter the number of your choice.

1<CR>

Index	Oxide	Amount of source - Kilograms
1	Al2O3	9310.4736
2	B2O3	54237.2070
3	CaO	1245.7639
4	Cr2O3	0.0000
5	FeO	70389.8125
6	K2O	23188.3066
7	Li2O	18373.4180
8	MgO	5362.8267
9	MnO	5908.4297
10	Na2O	65175.6172
11	NiO	251.1101
12	P2O5	13971.8291
13	SO3	0.0000
14	SiO2	259360.8594
15	ThO2	17575.0117
16	TiO2	4878.9946
17	UO2	2001.9706
18	ZrO2	0.0000
19	Residual	218.5536

Press RETURN to continue

<CR>

EXAMPLE 2

Example 2 deals with Option 2, checking an existing slurry for corrections. The user inputs required for this example are the file names (in order of use):

WPSOTEST2.DAT- file containing oxide weight percentages of slurry;

WPTOTEST2.DAT- file containing composition of tank contents;

WPOTEST2.DAT - file containing additives identification and composition.

Also required is the weight of the slurry in kilograms (10000), the weight of the tank contents (10000 kg), and the specific gravity of the desired glass (1.0).

PART A: Entering the Slurry Composition

Executing RECIPE using option 2 is very similiar to executing RECIPE using option 1. Once the command to run RECIPE has been entered and the menu appears, the number 2 is entered.

```
$ RUN RECIPE<CR>
```

```
*********************************************************
*                                                       *
*   Welcome to RECIPE Version 1.0                       *
*                                                       *
*   Would you like to:                                  *
*    1) Compose a new slurry or shim                    *
*    2) Check existing slurry for corrections           *
*    3) Terminate the program                           *
*                                                       *
*   Please enter the number of your selection.          *
*                                                       *
*********************************************************
```

```
2<CR>
```

The user is then requested to enter the name of the file which contains the composition of the the existing slurry.

```
Enter file containing composition of slurry:
```

Just as in example 1, if the file name is incorrect and does not exist in the current directory, an error message is displayed and the user is reprompted.

```
WPSOTERT2.DAT<CR>

>>> ERROR Opening WPSOTERT2.DAT

Enter file containing composition of slurry:
```

If the file name is incorrect but does exist in the directory, the program will attempt to read the file. In this case, the file has the incorrect format, so RECIPE will print an error message and then stop executing.

```
WPSOTERT2.DAT<CR>

>>> ERROR Detected in WPSOTERT2.DAT
>>> Incorrect File Format

FORTRAN STOP
$
```

Once again, the user must check the input and restart the program, making sure to enter the correct file name.

```
Enter file containing composition of slurry:

WPSOTEST2.DAT<CR>
```

Because this is the correct file, RECIPE will read the file and then display the contents for the user to verify.

```
Slurry Composition File:  WPSOTEST2.DAT

Al203           0.6416
B203            2.7874
CaO             0.1694
Cr203           0.0230
FeO             3.6447
K20             0.6801
Li20            0.2615
MgO             0.4178
```

MnO	0.3293
Na2O	2.3695
NiO	0.1553
P2O5	0.4070
SO3	0.0590
SiO2	6.0459
ThO2	1.4546
TiO2	0.3023
UO2	0.0725
ZrO2	0.0331
Residual	0.3023
CO3	1.1000
COOH	1.2000
NO3	1.4000

Is this the correct file? Y or N:

Y<CR>

After responding affirmatively, the user is asked for the weight of the slurry and once again asked to confirm his/her input.

Enter weight of slurry:
10000.<CR>

Slurry weight is 10000.0000 kilograms. Is this correct? Y or N:
Y<CR>

PART B: Entering the Tank Composition

Next, the program requests the name of the file containing the tank composition. The correct file name is WPTOTEST2.DAT. The behavior of this portion of the code is identical to that in PART A.

Enter file containing composition of tank contents:

WPTOTEST2.DAT

Tank File: WPTOTEST2.DAT

```
Table of Tank Composition
Units are Weight Percent

Oxide/Anion        Tank
   Names       Composition

Al2O3             0.2619
B2O3              0.6541
CaO               0.1913
Cr2O3             0.0484
FeO               0.6957
K2O               0.2995
Li2O              0.2816
MgO               0.1300
MnO               0.1599
Na2O              0.8257
NiO               0.0899
P2O5              0.2264
SO3               0.0519
SiO2              0.3361
ThO2              0.2236
TiO2              0.4309
UO2               0.2913
ZrO2              0.2883
Residual          0.2586
CO3               1.1000
COOH              1.2000
NO3               1.4000

Is this the correct file? Y or N:
Y<CR>

Enter weight of tank contents (in kilograms):
10000.<CR>

Weight of tank contents is   10000.0000 kilograms.
Is this correct? Y or N:
Y<CR>
```

PART C: Entering the Desired Glass Composition

The last file name requested by RECIPE is of the file containing the desired composition of the resulting glass. Once again, the file contents

are displayed, and the user is asked to verify the contents. The user is also asked to supply and verify the specific gravity.

```
Enter file containing desired glass composition and tolerances:

WPOTEST2.DAT<CR>

Glass Composition File:  WPOTEST2.DAT

Desired Glass Tolerances and Composition Table
          Units are Weight Percent

Oxide        Minimum Acceptable    Target    Maximum Acceptable
Al203            2.0000           2.8300          4.0000
B203             9.0000           9.9500         12.0000
CaO              0.0000           0.6000          2.0000
Cr203            0.0000           0.3100          0.5000
FeO             10.0000          12.1600         14.0000
K20              2.5000           3.5700          4.5000
Li20             2.0000           3.0300          3.5000
MgO              0.5000           1.3000          2.0000
MnO              0.0000           1.3100          2.0000
Na20             9.0000          10.9400         13.0000
NiO              0.0000           0.3400          1.0000
P205             0.0000           2.5100          3.0000
S03              0.0000           0.2200          0.5000
Si02            41.0000          44.8800         48.0000
Th02             0.0000           3.5800          4.5000
Ti02             0.0000           0.9800          4.0000
U02              0.0000           0.5600          3.0000
Zr02             0.0000           0.2900          3.0000
Residual         0.0000           0.6400          3.0000

Is this the correct file? Y or N:
Y<CR>

Enter the specific gravity of the desired glass:
1.0<CR>
```

The specific gravity of the desired glass is 1.0000.

Is this correct? Y or N:

Y<CR>

PART D: Continuing with RECIPE

RECIPE will now display the results of its calculations in tabular form. After each table is displayed, the program will wait for the user to press RETURN before continuing.

Table of Glass Tolerances and Calculated Glass Composition
(Units in Weight Percent)

Note: If any of the X(j)'s have been set to zero, the resulting glass will deviate from the desired glass composition.

Oxide	Minimum Tolerance	*WPG or WPGO	Desired Glass	Maximum Tolerance	In/Out Bounds
Al203	2.0000	3.4881	2.8300	4.0000	OK
B203	9.0000	13.2871	9.9500	12.0000	OUT OF BOUNDS
CaO	0.0000	1.3926	0.6000	2.0000	OK
Cr203	0.0000	0.2757	0.3100	0.5000	OK
FeO	10.0000	16.7572	12.1600	14.0000	OUT OF BOUNDS
K20	2.5000	3.7821	3.5700	4.5000	OK
Li20	2.0000	2.0968	3.0300	3.5000	OK
MgO	0.5000	2.1153	1.3000	2.0000	OUT OF BOUNDS
MnO	0.0000	1.8885	1.3100	2.0000	OK
Na20	9.0000	12.3361	10.9400	13.0000	OK
NiO	0.0000	0.9466	0.3400	1.0000	OK
P205	0.0000	2.4453	2.5100	3.0000	OK
S03	0.0000	0.4282	0.2200	0.5000	OK
Si02	41.0000	24.6399	44.8800	48.0000	OUT OF BOUNDS
Th02	0.0000	6.4791	3.5800	4.5000	OUT OF BOUNDS
Ti02	0.0000	2.8308	0.9800	4.0000	OK
U02	0.0000	1.4046	0.5600	3.0000	OK
Zr02	0.0000	1.2408	0.2900	3.0000	OK
Residual	0.0000	2.1653	0.6400	3.0000	OK

*Computed glass composition after mixing additives with the tank and assuming no losses from off gas.

Press RETURN to continue

<CR>

At this point, the user is given the option whether or not to save the oxide weight percentages of the mixture that would be obtained by combining the contents of the tank with the slurry preparation.

```
Do you want to:
  1) Write WPMO to a file and return to the main menu
  2) Return to the main menu without saving WPMO

Please enter the number of your choice (1 or 2)
```

If the user chooses to save the information, the number 1 should be entered,

```
1<CR>
```

RECIPE will respond by supplying the name of the newly created file and then returning the user to the main menu.

```
The WPMO array has been saved in the file, WPMO_30_JUL_87_15_00.DAT

You are now being returned to the main menu.
```

```
*******************************************************
*                                                     *
*   Welcome to RECIPE Version 1.0                     *
*                                                     *
*   Would you like to:                                *
*    1) Compose a new slurry or shim                  *
*    2) Check existing slurry for corrections         *
*    3) Terminate the program                         *
*                                                     *
*   Please enter the number of your selection.        *
*                                                     *
*******************************************************
```

If the user does not wish to save the information, the number 2 should be entered and the program will simply return the user to the main menu.

```
Do you want to:
  1) Write WPMO to a file and return to the main menu
  2) Return to the main menu without saving WPMO

Please enter the number of your choice (1 or 2)

2<CR>

You are now being returned to the main menu.
```

```
*********************************************************
*                                                       *
*   Welcome to RECIPE Version 1.0                       *
*                                                       *
*   Would you like to:                                  *
*    1) Compose a new slurry or shim                    *
*    2) Check existing slurry for corrections           *
*    3) Terminate the program                           *
*                                                       *
*   Please enter the number of your selection.          *
*                                                       *
*********************************************************
```

Once the menu is displayed, the program can be exited by entering the number 3. The user will then be returned to the operating system.

```
3<CR>
$
```

6.0 TEST CASES

The test cases are found on the microfiche cards on the inside back cover of this report. The procedures and files used for each of the four test runs of the RECIPE Version 1.0 software are described. Each run test is either (a) the calculations that are used to compose a new slurry or (b) the calculations to determine whether the composition of an existing cold slurry will produce a glass of the correct composition when mixed with the batch of the waste.

7.0 LIST OF RECIPE MODULES

This section of the user's guide lists the subroutine and functions in alphabetical order (following the main program). A brief description of the module is given along with a list of limits and/or prerequisites (if any). At the end of this section is a list of modules as they are called for in both option 1 and option 2.

In addition to the modules, RECIPE also utilizes a file titled PARAMETER.PRM. This file contains the parameters used by the RECIPE code and is included into the program by a FORTRAN statement. In order to change the parameters that are used by the code, they must be changed by editing this file. The current contents of the file follow:

```
PARAMETER (MAX_OXIDES = 100,MAX_ADDITIVES = 5).
```

MODULE DESCRIPTIONS

Program RECIPE

Purpose

RECIPE Version 1.0 is designed to allow the user to determine the necessary weights of oxide-providing additives to be added to a THOREX-zeolite waste mix of known composition in order that resulting mixture will form a glass that meets a target glass composition within specified tolerances.

RECIPE Version 1.0 also tests to see whether an already composed slurry or shim will, when mixed with the THOREX-zerolite waste mix of known composition, result in the target composition within specified tolerances.

Limitations and Prerequisites

This program requires the use of strictly structured input data files. The data files and there structures are currently described in FILES.DOCUMENTATION. To run, this program must be linked with the LINPACK/BLAS libraries.

Routines Used: (In order of use)

MAIN_MENU, OPEN_OUTPUT, WPA_FILL, RP_FILL, WPTO_FILL, WPO_FILL, WP_CALC, WP_SAVE, PIVOT_CALC, MODIFY_WP, WP_SAVE, EXACT_SOLUTION, NEGATIVE_CHECK, NEG_OPTIONS, SOURCE_CALC, ADDITIVE_CALC, WPG_CALC, ANION_RATIOS, CHECK_TOLERANCE, PRINT_TABLE, WPSO_FILL, SAVE_WPMO

Subroutine ADDITIVE_CALC

Purpose

This subroutine calculates and displays the amount of each additive to be added to the slurry for each source. The information is also written to the general output file.

Limitations and Prerequisites

The subroutine SOURCE_CALC must be called before the call to ADDITIVE_CALC.

Subroutine ANION_RATIOS

Purpose

This subroutine calculates the ratio of anion mass to glass mass, the ratio of anion mass to tank mass (after addition of the slurry), and the volume of glass that would be produced if the slurry was mixed with the contents of the tank.

Limitations and Prerequisites

The subroutine WPG_CALC must be called before ANION_RATIOS in order for the argument MGLASS to contain the mass of the glass.

The argument NUM_OXIDES must contain the number of oxides.

The argument NUM_ANIONS must contain the number of anions.

Subroutine CALC_MASS

Purpose

Calculates mass of oxides in the glass that would be formed if the additives were added to the tank.

Limitations and Prerequisites

None.

Subroutine CHECK TOLERANCE

Purpose

This subroutine compares [WPG] with the minimum and maximum tolerances to determine whether the glass would be acceptable.

Limitations and Prerequisites

The argument NUM_OXIDES must contain the number of oxides.

Subroutine DISPLAY OPTIONS

Purpose

This subroutine displays user options and returns their selection. It also writes user selection to the general output file.

Limitations and Prerequisites

None.

Subroutine EXACT SOLUTION

Purpose

This subroutine solves for the exact solutions X in the system transpose(WP) * X = TARGET. Calculations are in double precision, except those performed by DDSDOT, which are in REAL*16 precision. If the initial solution has lost more than four significant digits (arbitrary criterion), the subroutine attempts to improve numerical significance by iterative improvement, following the procedure on page 1.9 of the LINPACK manual (SUBROUTINE SGEIM).

Limitations and Prerequisites

The WP array must represent a square matrix of order N.

The current capacity is N <= 32. To redefine it, set PARAMETER LDB to desired value. Argument LDA need not be the same as LDB.

For definitions of "condition number" and similar measures for determining that a matrix is "singular," refer to the LINPACK User's manual, pp I.7-I.10 and Chapter 1.

The DGESL input argument JOB causes the routine to obtain solutions to either transpose(WP) * X = TARGET (JOB=1) or WP * X = TARGET (JOB=0). Currently, JOB is set to 1. This setting impacts iterative improvement.

REAL*8 FUNCTION DDSDOT

PURPOSE:

To calculate the difference between the dot product of two vectors X and Y and a scalar C. All inputs are in double precision, and all calculations are performed to "quad" or REAL*16 precision.

The function returns itself set to (X dot Y) - C.

If N le 0 the function returns itself set to -C.

Subroutine MAIN MENU

Purpose

This subroutine displays the main menu selections for the user and reads his/her choice.

Limitations and Prerequisites

None.

Subroutine MODIFY WP

Purpose

This routine replaces the values for the pivot oxide in the WP matrix with the oxide values from the tank composition. In effect, it is modifying the WP matrix so that the tank contents composition acts as a source for one of the oxides.

Limitations and Prerequisites

MODIFY WP must be called after the WP matrix has been created and PIVOT_CALC has been called.

Subroutine NEGATIVE_CHECK

Purpose

This subroutine checks the exact solution array [X] for negative values. If a negative value is found, NEGATIVE_CHECK then searches to determine whether the negative value occurred for the pivot oxide.

Limitations and Prerequisites

The actual parameter, NUM_OXIDES, must contain the identical number of elements as the array [X], not necessarily the same as the length.

The actual parameter, PIVOT_OXIDE, must contain the index number of the array [X] which corresponds to the pivot oxide.

Subroutine NEG_OPTIONS

Purpose

This subroutine informs the user whether the exact solution contains negative values. If the pivot oxide is not negative, then the user may either zero negative values for all oxides or choose which negative values are to be zeroed. If the pivot oxide is negative, the user is warned to check the inputs and forced to restart.

Limitations and Prerequisites

Subroutine NEGATIVE_CHECK must be called before NEG_OPTIONS may be called.

The number of elements of [X] and [COMPONENT_NAMES] must be contained in the argument NUM_OXIDES.

Subroutine OPEN_FILE

Purpose

This subroutine opens the various input files.

Limitations and Prerequisites

None.

Subroutine OPEN OUTPUT

Purpose

This subroutine names and opens the output file if the user decides to compose a new slurry or check an existing slurry.

Limitations and Prerequisites

OPEN OUTPUT is called by MAIN TEST.

Subroutine PIVOT CALC

Purpose

This subroutine computes the pivot oxide.

Limitations and Prerequisites

PIVOT CALC must not be called until the following routines have been called WPA_FILL,RP_FILL,WPO_FILL,WPTO_FILL.

Subroutine PRINT NEG X

Purpose

This subroutine displays the negative exact solution values.

Limitations and Prerequisites

The argument NUM_OXIDES must contain the number of array elements of arrays [X], [COMPONENT_NAMES].

Subroutine PRINT TABLE

Purpose

This subroutine prints summary information in tabular form for the user.

Limitations and Prerequisites

Either subroutines WPGO_CALC and CHECK_TOLERANCE must be called and the argument FIRST set to FALSE, or the subroutines WPG_CALC, ANION_RATIOS, and CHECK_TOLERANCE must be called and the argument FIRST set to TRUE.

Subroutine PRINT X

Purpose

This subroutine displays the exact solution values.

Limitations and Prerequisites

The argument NUM_OXIDES must contain the number of array elements of arrays [X], [COMPONENT_NAMES].

Subroutine READ SOURCE

Purpose

This subroutine reads the data for one oxide source.

Limitations and Prerequisites

READ SOURCE is called by WPA_FILL.

Subroutine SAVE WPMO

Purpose

Use of this subroutine gives the user the option to save the [WPMO]. If the user chooses to save the [WPMO], a file is created with a unique name. The COMPONENT_NAMES and the corresponding [WPMO] values are written to that file. The unique file name has the form, WPMO_DD_MMM_YY_HH_MM.DAT, where DD_MMM_YY is the system date (day, month,year) and HH_MM is the system time (hour,minute).

Limitations and Prerequisites

None.

Subroutine SELECT OXIDES

Purpose

This subroutine asks the user how many negative exact solution values are to be zeroed and then asks for the index(s) of those values. All user choices are written to the general output file.

Limitations and Prerequisites

The argument NUM_OXIDES must contain the number of array elements of arrays [X], [COMPONENT_NAMES], and [SELECT].

Subroutine SOURCE_CALC

Purpose

This subroutine calculates the amount of each source to be added to the tank.

Limitations and Prerequisites

The number of oxides must be contained in the argument NUM_OXIDES. The weight of the tank must be contained in the argument WTO. The index of the pivot oxide must be contained in the argument PIVOT_OXIDE.

Real Function TOTAL-MASS

Purpose

Calculates the total mass of all the oxides, given an array whose elements contain the individual oxide masses.

Limitations and Prerequisites

The argument NUM_OXIDES must contain the number of oxides. The subroutine CALC_MASS must be called before TOTAL_MASS in order to set the array [M], which contains the individual oxide masses.

Subroutine WPA_FILL

Purpose

This subroutine reads in the oxide additives (WPA) matrix.

Limitations and Prerequisites

None.

Subroutine WPG CALC

Purpose

This subroutine calculates the composition of the glass that would result if the additives were put in the tank and the material heated to form the oxides.

Limitations and Prerequisites

None.

Subroutine WPMO CALC

Purpose

This subroutine calculates the individual oxide (or anion) masses. It also calculates the sum of all masses.

Limitations and Prerequisites

The argument NUM_OXIDES must contain the number of oxides.

Subroutine WPO FILL

Purpose

This subroutine read data from the WPO, target composition and tolerances, and data file.

Limitations and Prerequisites

WPO FILL must be called after WPA_FILL or WPSO_FILL.

Subroutine WPSO FILL

Purpose

This subroutine reads in the slurry composition (WPSO) matrix.

Limitations and Prerequisites

None.

Subroutine WP_CALC

Purpose

This subroutine reads in data from the tank contents file.

Limitations and Prerequisites

WP_CALC Must be called after WPA_FILL or WPSO_FILL.

Subroutine WP SAVE

Purpose

This subroutine computes the weight percents of the oxides, given the relative percentages of the additives and the weight percents of the additives.

Limitations and Prerequisites

This subroutine must be called after WPA_FILL and RP_FILL.

Subroutine WP SAVE

Purpose

This subroutine writes the WP matrix to the output file.

Limitations and Prerequisites

WP_SAVE must be called after the WP array has been calculated.

Subroutine ZERO X

Purpose

This subroutine zeroes the appropriate negative values of [X].

Limitations and Prerequisites

The argument NUM_OXIDES must contain the number of elements of arrays [X] and [SELECT]. The subroutine SELECT_OXIDES must be called before ZERO_X.

****

LIST OF MODULES IN ORDER OF CALLING

Option #1

1. MAIN_MENU
2. OPEN_OUTPUT
3. WPA_FILL
   A) OPEN_FILE
4. RP_FILL
   A) OPEN_FILE
5. WPTO_FILL
   A) OPEN_FILE
6. WPO_FILL
   A) OPEN_FILE
7. WP_CALC
8. WP_SAVE
9. EXACT_SOLUTION
10. PRINT_X
11. NEGATIVE_CHECK
12. NEG_OPTIONS
    A) DISPLAY_OPTIONS*
    B) SELECT_OXIDES*
    C) ZERO_X*
13. SOURCE_CALC
14. ADDITIVE_CALC
15. WPG_CALC
    A) CALC_MASS
    B) TOTAL_MASS
16. ANION_RATIOS
17. CHECK_TOLERANCE
18. PRINT_TABLE

Option #2

1. MAIN_MENU
2. OPEN_OUTPUT
3. WPSO_FILL
   A) OPEN_FILE
4. WPTO_FILL
   A) OPEN_FILE
5. WPO_FILL
   A) OPEN_FILE
6. WPGO_CALC
7. CHECK_TOLERANCE
8. PRINT_TABLE
9. SAVE_WPMO**

*These modules are only called if negative values occur in the exact solution.
**This module is only called if the user elects to save the WPMO.

DISTRIBUTION

No. of Copies

OFFSITE

10 DOE Office of Scientific and Technical Information

2 Geologic Repository Division
DOE Office of Civilian Radioactive Waste Management
Forrestal Building
Washington, DC 20575
ATTN: J. R. Hilley, RW-30
S. Kale

3 DOE Office of Defense Waste & Byproducts Managements
GTN
Washington, DC 20545
ATTN: T. C. Chee, DP-123
G. H. Daly, DP-124
J. E. Lytle, DP-12

3 DOE Office of Terminal Waste Disposal & Remedial Action
GTN
Washington, DC 20545
ATTN: J. A. Coleman, NE-25
T. W. McIntosh, NE-25
H. F. Walter, NE-25

A. T. Clark
Division of Fuel Material Safety
Nuclear Regulatory Commission
Washington, DC 20555

V. Stello
Office of the Executive Director for Operations
Mail Station 6209
Nuclear Regulatory Commission
Washington, DC 20555

G. L. Sjoblom
Environmental Protection Agency
Office of Radiation Programs
401 M Street, S.W.
Washington, DC 20460

No. of Copies

J. M. McGough
DOE Albuquerque Operations Office
P.O. Box 5400
Albuquerque, NM 87185

P. G. Hagan
Joint Integration Office
Carlmont Executive
4308 Carlisle N. E.
Albuquerque, NM 87107

E. Maestas
DOE West Valley Operations Office
P.O. Box 191
West Valley, NY 14171

J. P. Hamric
DOE Idaho Operations Office
550 Second Street
Idaho Falls, ID 83401

F. T. Fong
DOE San Francisco Operations
133 Broadway
Oakland, CA 94612

M. R. Jugan
DOE Oak Ridge Operations Office
P.O. Box E
Oak Ridge, TN 37830

W. H. Brumley
DOE Savannah River Operations Office
P.O. Box A
Aiken, SC 29801

M. J. Steindler
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439

No. of Copies

C. S. Abrams
Argonne National Laboratory
P.O. Box 2528
Idaho Falls, ID 83401

3 Battelle Memorial Institute
Project Management Division
505 King Avenue
Columbus, OH 43201
ATTN: W. A. Carbeiner
W. S. Madia
Technical Library

L. D. Ramspott
Lawrence Livermore National Laboratory
University of California
P.O. Box 808
Livermore, CA 94550

D. T. Oakley, MS 671
Los Alamos Scientific Laboratory
P.O. Box 1663
Los Alamos, NM 87544

4 Oak Ridge National Laboratory
P.O. Box Y
Oak Ridge, TN 37830
ATTN: J. O. Blomeke
D. W. Turner

Sandia Laboratories
Technical Library
P.O. Box 5800
Albuquerque, NM 87185

J. R. Berreth
Westinghouse Idaho Nuclear Co., Inc.
P.O. Box 4000
Idaho Falls, ID 83401

No. of Copies

6 E. I. du Pont de Nemours Company
Savannah River Laboratory
Aiken, SC 29801
ATTN: R. G. Baxter
M. D. Boersma
J. G. Glasscock
J. R. Knight
M. J. Plodinec
C. T. Randall

R. Shaw
Electric Power Research Institute
3412 Hillview Avenue
P.O. Box 10412
Palo Alto, CA 94304

5 West Valley Nuclear Services Company
P.O. Box 191
West Valley, NY 14171
ATTN: S. M. Barnes
J. E. Krausss
S. J. Marchette
J. M. Pope

ONSITE

5 DOE Richland Operations Office

E. A. Bracken
C. E. Collantes
J. R. Hunter
J. J. Sutey
O. L. Olson

9 Westinghouse Hanford Company

J. M. Henderson
R. E. Lerch
R. D. Prosser
J. L. Scott
R. A. Sexton

No. of Copies

Westinghouse Hanford Company (contd)

J. D. Watrous
D. D. Wodrich
B. A. Wolfe
File Copy

66 Pacific Northwest Laboratory

C. R. Allen
W. W. Ballard, Jr.
D. N. Berger
W. F. Bonner
B. B. Brenden
R. A. Brouns (2)
J. L. Buelt
H. C. Burkholder
D. G. Coles
R. D. Dierks
T. A. Erb (2)
D. W. Faletti (10)
R. W. Goles
H. Harty
W. O. Heath
L. K. Holton
R. S. Kemper
S. S. Koegler

No. of Copies

Pacific Northwest Laboratory (contd)

D. E. Knowlton
W. L. Kuhn
L. T. Lakey
D. E. Larson
S. F. Liebetrau
C. A. LoPresti
J. L. McElroy
K. H. Oma
R. K. Nakoaka
J. M. Perez, Jr.
M. E. Peterson
W. A. Ross
P. A. Scott
J. M. Seay
G. H. Sevigny
D. H. Siemens
J. L. Straalsund
R. E. Thornhill
R. L. Treat
J. H. Westsik, Jr. (10)
Publishing Coordination (2)
Technical Report Files (5)