Visual Basic Conversion Tool

Over the course of the semester, I have endeavored to create a useful conversion tool that would aid scientists in testing models of H2O and CO2 solubility in a silicate melt. Creating these models is extremely important in understanding the petrogenesis of mantle-derived melts. A tool for testing these models will allow scientists to more easily constrain model parameters in order to easily perfect these models and make them more accurate.

In the creation of a solubility model, analyses are taken of melt inclusions in order to determine the mole fraction of H2O and CO2 dissolved in the melt. Additional bulk analyses of the rock determine the weight percent of the component oxides in the entire rock. However, a rock at Earth's surface does not contain much, if any, H2O or CO2, since the solubility of these components increases with depth. So, the bulk measurements taken are essentially anhydrous values, which do not account for H2O and CO2. In order to obtain a useful analysis of a hydrous rock at depth, these two analyses must be combined.

This is not a simple conversion, since the two data sets are in terms of different units: mole fraction and weight percent. An MS Excel-based, easy to use program that could perform these calculations is useful in looking at meaningful data of hydrous silicate melts.

Initially, in order to learn the basics of the Visual Basic coding language, a simple conversion tool, titled WtpertoMolper, was created that would simply convert given weight percent values into mole percent values. I will take you through a brief tutorial of this program to give you an idea of how this program works and how it was useful in constructing the final product, Conversion-tool.

Operating WtpertoMolper:

Open the file.

Using the worksheet tabs at the bottom of the spreadsheet, bring the "WttoMol" sheet to the front (if it's not already there). Under column B, labeled *Wt. %*, input values for a bulk analysis of a rock in terms of weight percent. The total at the bottom of this column will confirm if these values have been normalized (add to 100), or if they still have measurement error (do not add to 100). It is okay if your values are not normalized, the conversion process will normalize them for you when you convert to mole percent.

Once your values have been entered, click the blue button labeled "Convert weight percent to mol percent". The program will run through the calculations and output the mole percent values under column D, labeled *Mol %*. These calculations can be repeated indefinitely with as many compositions as needed. The green "Clear Output" button clears cells D2 to D15 (the output Mol % values). The yellow "Clear Input" button clears cells B2 to B15 (the user input Wt. % values).

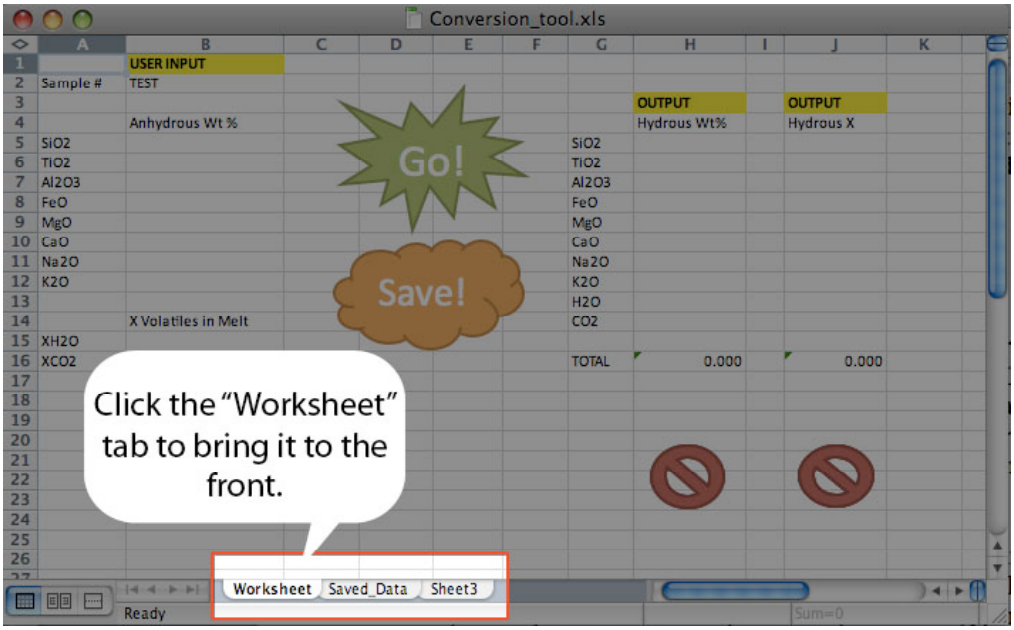
The WtpertoMolper was a very simple code that helped me to learn the essentials of the Visual Basic coding language and aided in the creation of the final

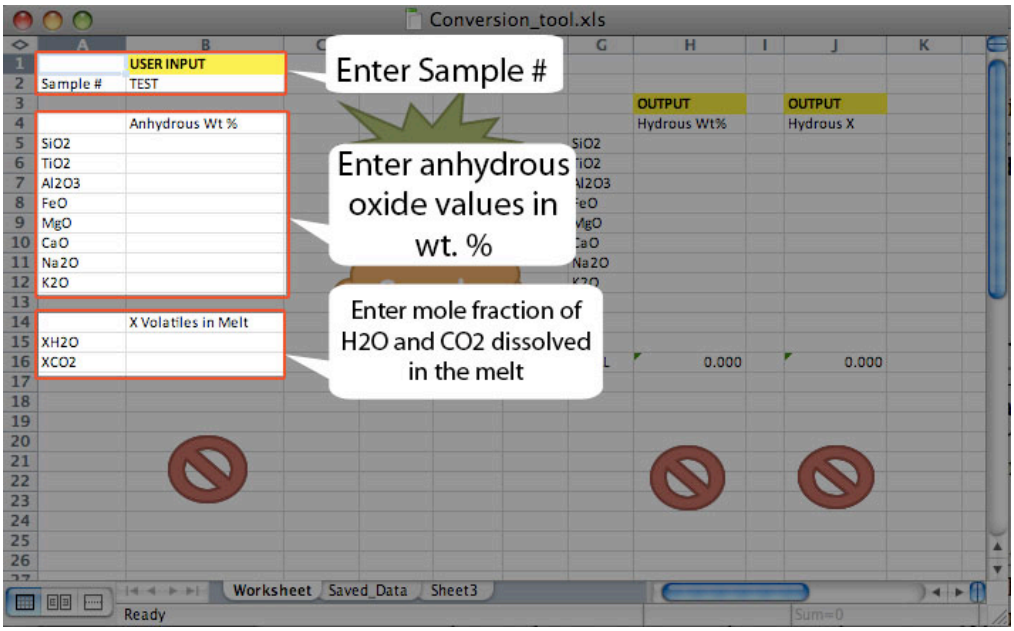
project, Conversion-Tool. I will take you through a tutorial of this program, and explain how novice to advanced users can use it.

Operating Conversion-tool:

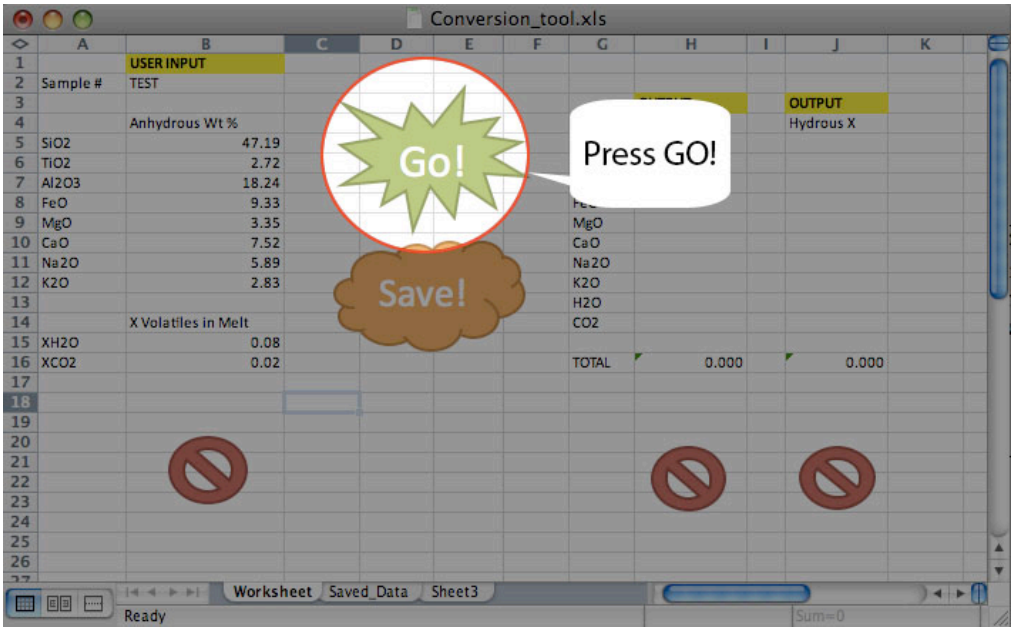
1. Open the file "Converstion_tool.xls"

2. Using the worksheet tabs at the bottom of the spreadsheet, bring the "Worksheet" sheet to the front (if it is not already there). Under Column B, labeled *User Input*, enter your sample number, anhydrous values for each oxide from bulk analysis of your rock in terms of weight percent (under *Anhydrous wt %*), and values for amount of H2O and CO2 dissolved in the melt in terms of mole fraction (under *X Volatiles in melt*).

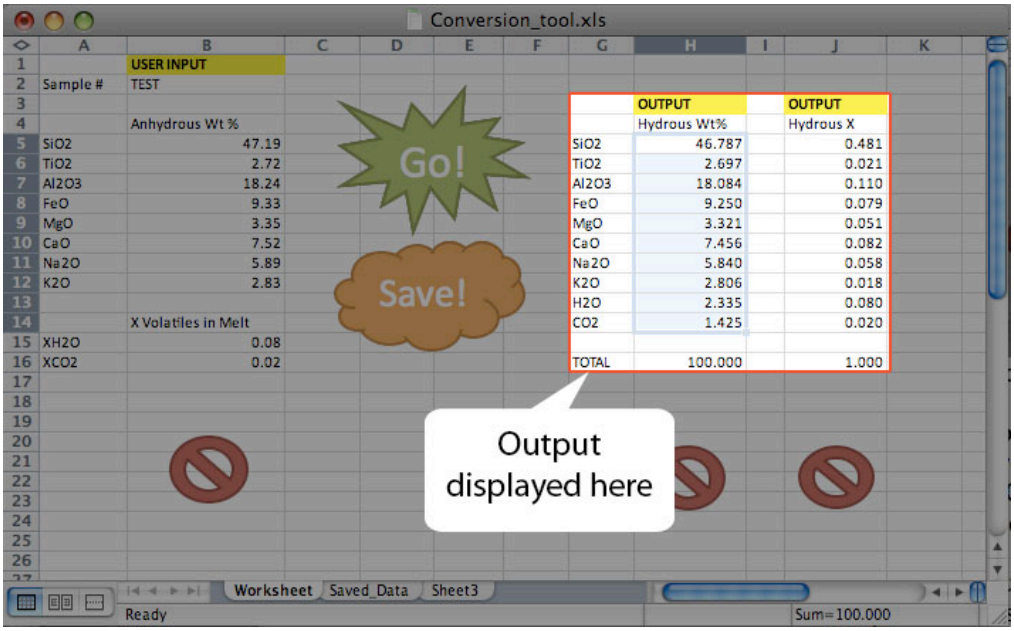




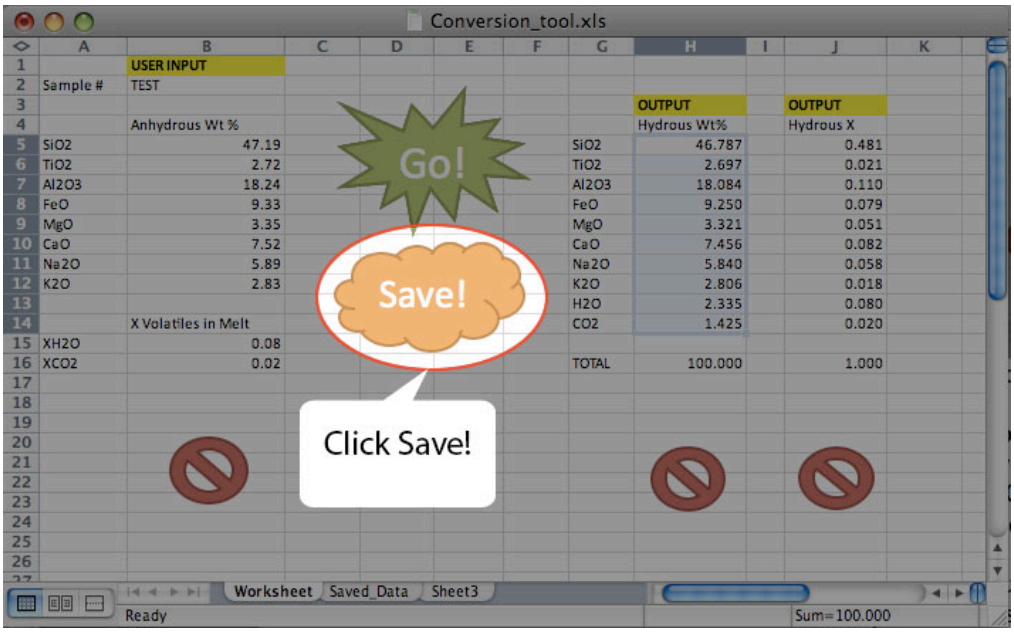
3. Click the big green “Go!” button to run the program. The computer will automatically calculate hydrous values for your rock in terms of both weight percent, under column H (labeled *Hydrous Wt. %*) and mole fraction under column J (labeled *Hydrous X*).



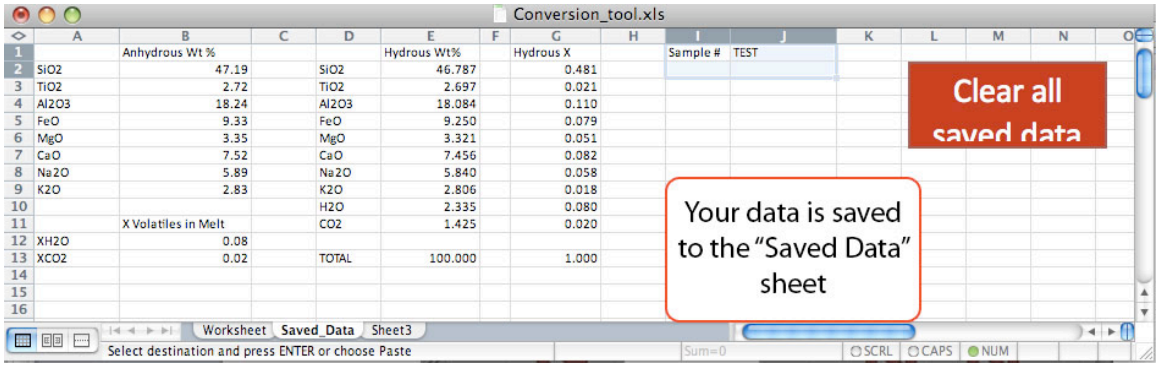
4. The totals underneath each output column should total to 100.00 for weight percent and 1.000 for mole fraction. This is a check to make sure that your calculations were run properly. Please note, this works best for mole fraction volatile values between 0 and 0.2 mole fraction. Error of the hydrous mole fraction calculation increases as mole fraction of H2O and CO2 increase, but the calculations have minimal error for values under 0.2 mole fraction.



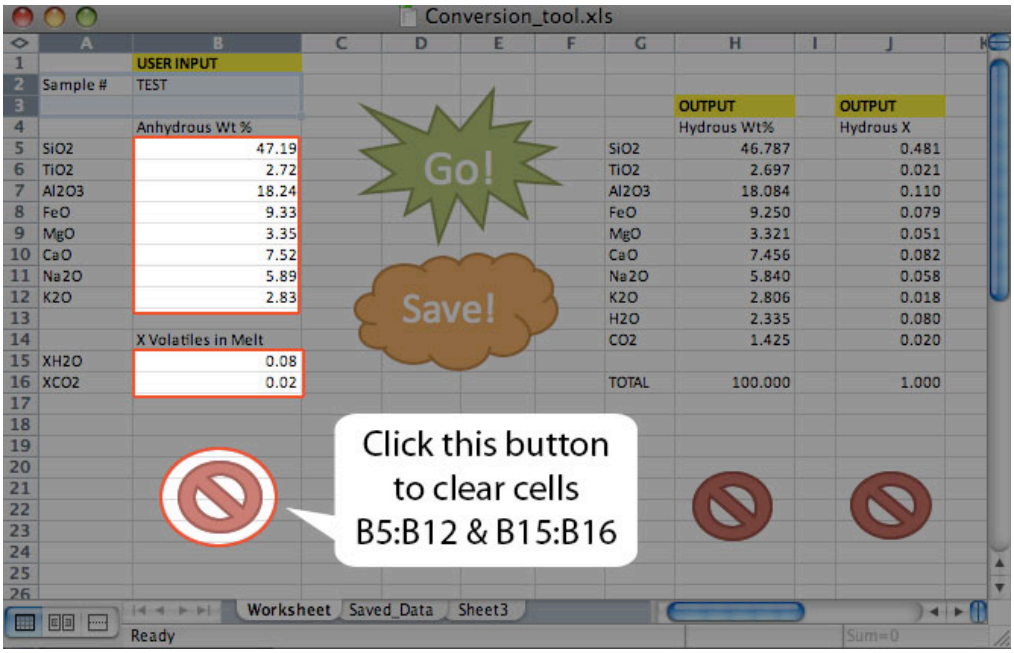
5. This program allows you to perform several calculations and look over all of the data from these calculations at once. If you wish to store the calculation you have just performed, click the big orange “Save!” button. This effectively saves the information from the currently displayed calculation and pastes it to the “Saved Data” worksheet. Upon clicking this button, you will be taken to the “Saved Data” worksheet, where you can see how your data is saved.



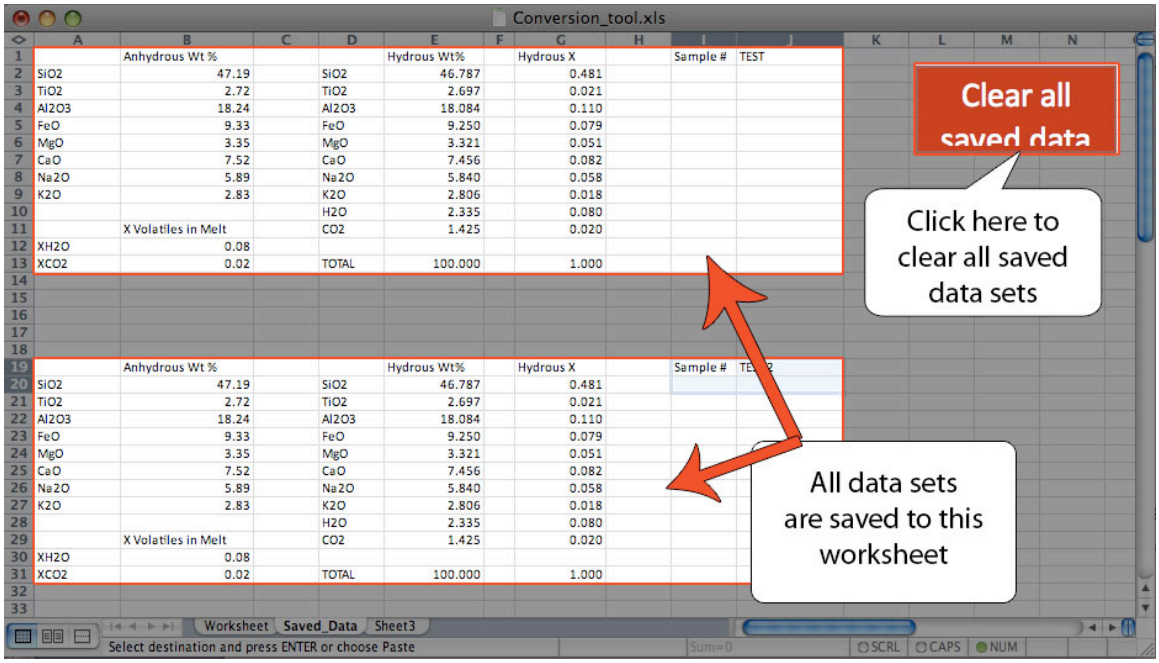
6. You may then run a new calculation and save it to your “Saved Data” worksheet. To do this, use the worksheet tabs at the bottom of the worksheet to go back to the “Worksheet” sheet. The red circle symbols below each column will clear values from its column. Click the red symbol below the *User Input* column to clear your previous calculation values. If you wish, you may also clear both *Output* columns using the red clear buttons, but your next calculation will simply overwrite any values in those columns.



7. Repeat steps 2-5 to perform a new calculation. Upon saving a second calculation (or a third, or a fourth…) you will notice that all of your calculations will be displayed on the “Saved Data” worksheet. This allows you to look at all of your data in one place.



8. When you are finished with your analyses, you may want to clear the "Saved Data" sheet for future use. To do this, tab to the "Saved Data" sheet and click the big red "Clear all saved data" button, located to the right of the first saved calculation.



The Code:

Now you know how to operate the excel spreadsheet Conversion-tool.xls. The following is a tour of the Visual Basic code explaining what the program is doing when and where and the math behind it all.

The beginning of the code is standard to all programs and is the place where we define all of the variables that will be used in any calculations the program will use henceforth.

```
' Conversion_tool Macro
' Program that converts anhydrous wt% oxides plus hydrous mole fraction or wt% H2O and CO2 into hydrous wt% or mole fraction oxides + volatiles
' Created by Kayla Iacovino for Gordon Moore and the Omni Pressure Lab, 2008

Option Explicit 'Forces declaration of all variables

'Declaration of global variables
Dim SiO2 As Single, TiO2 As Single, Al2O3 As Single, FeO As Single
Dim MgO As Single, CaO As Single, Na2O As Single, K2O As Single
Dim XH2Oin As Single, XCO2in As Single

Dim XSiO2a As Single, XTiO2a As Single, XAl2O3a As Single
Dim XFeOa As Single, XMgOa As Single, XCaOa As Single, XNa2Oa As Single
Dim XK2Oa As Single

Dim XSiO2hy As Single, XTiO2hy As Single, XAl2O3hy As Single, XFeOhy As Single
Dim XMgOhy As Single, XCaOhy As Single, XNa2Ohy As Single, XK2Ohy As Single

Dim XSiO2h As Single, XTiO2h As Single, XAl2O3h As Single, XFeOh As Single
Dim XMgOh As Single, XCaOh As Single, XNa2Oh As Single, XK2Oh As Single
Dim XH2Ofin As Single, XCO2fin As Single

Dim perSiO2h As Single, perTiO2h As Single, perAl2O3h As Single
Dim perFeOh As Single, perMgOh As Single, perCaOh As Single
Dim perNa2Oh As Single, perK2Oh As Single, perH2Oh As Single
Dim perCO2h As Single

Dim sheet As Object
Dim savesheet As Object

'Declaration of Constants

Const MWSiO2 = 60.08
Const MWTiO2 = 79.88
Const MWAl2O3 = 101.96
Const MWFeO = 71.85
Const MWMgO = 40.3
Const MWCaO = 56.08
Const MWNa2O = 61.98
Const MWK2O = 94.2
Const MWH2O = 18.02
Const MWCO2 = 44.01
```

The next section, Sub conversion( ), initializes all of our variables and contains the operations which the program will run through in a specific order. All variables are listed, and all of the subroutines are listed in order of operation. Each subroutine is individually defined in the remainder of the code. Sub conversion ( ) pulls the information from each subroutine and performs them in the correct order when the program runs.

```
'Get input data
GetEntryData SiO2, TiO2, Al2O3, FeO, MgO, CaO, Na2O, K2O, _
XH2Oin, XCO2in

'Calculate anhydrous mole fraction values
CalcXAnh SiO2, TiO2, Al2O3, FeO, MgO, CaO, Na2O, K2O, _
XSiO2a, XTiO2a, XAl2O3a, XFeOa, XMgOa, XCaOa, XNa2Oa, XK2Oa

'Calculate hydrous mole fraction values
CalcXhydrous XSiO2a, XTiO2a, XAl2O3a, XFeOa, XMgOa, XCaOa, _
XNa2Oa, XK2Oa, XH2Oin, XCO2in, XSiO2hy, XTiO2hy, XAl2O3hy, XFeOhy, XMgOhy, _
XCaOhy, XNa2Ohy, XK2Ohy

'Normalize hydrous mole fraction values (corrects for error)
NormXhydrous XSiO2hy, XTiO2hy, XAl2O3hy, XFeOhy, XMgOhy, XCaOhy, _
XNa2Ohy, XK2Ohy, XH2Oin, XCO2in, XSiO2h, XTiO2h, XAl2O3h, XFeOh, XMgOh, XCaOh, _
XNa2Oh, XK2Oh

'Calculate hydrous wt% values
Calcwtperhydrous XSiO2h, XTiO2h, XAl2O3h, XFeOh, XMgOh, XCaOh, _
XNa2Oh, XK2Oh, XH2Oin, XCO2in, perSiO2h, perTiO2h, perAl2O3h, _
perFeOh, perMgOh, perCaOh, perNa2Oh, perK2Oh, perH2Oh, perCO2h

'Output X values
SendoutputX XSiO2h, XTiO2h, XAl2O3h, XFeOh, XMgOh, XCaOh, _
XNa2Oh, XK2Oh, XH2Oin, XCO2in

'Output Wt% values
Sendoutputwtper perSiO2h, perTiO2h, perAl2O3h, _
perFeOh, perMgOh, perCaOh, perNa2Oh, perK2Oh, perH2Oh, perCO2h
```

The first subroutine is *Get entry data*. This simply acquires all of the values input by the user on the excel spreadsheet and stores them.

```
'Get input data
Sub GetEntryData(SiO2anh As Single, TiO2anh As Single, Al2O3anh As Single, _
            FeOanh As Single, MgOanh As Single, CaOanh As Single, _
            Na2Oanh As Single, K2Oanh As Single, XH2Oinit As Single, _
            XCO2init As Single)

  Set sheet = Application.Workbooks("Conversion_tool.xls").Worksheets("Worksheet")

  SiO2anh = sheet.Range("SiO2in").Value
  TiO2anh = sheet.Range("TiO2in").Value
  Al2O3anh = sheet.Range("Al2O3in").Value
  FeOanh = sheet.Range("FeOin").Value
  MgOanh = sheet.Range("MgOin").Value
  CaOanh = sheet.Range("CaOin").Value
  Na2Oanh = sheet.Range("Na2Oin").Value
  K2Oanh = sheet.Range("K2Oin").Value
  XH2Oinit = sheet.Range("XH2Oin").Value
  XCO2init = sheet.Range("XCO2in").Value
```

Next is *CalcXAnh*. This takes the users input anhydrous oxide values (input in terms of weight percent) and converts them to units of mole fraction. Function *mol_prop_ox* defines the mathematical function used in the conversion.

```
Sub CalcXAnh(SiO2wp As Single, TiO2wp As Single, Al2O3wp As Single, FeOwp As Single, _
        MgOwp As Single, CaOwp As Single, Na2Owp As Single, K2Owp As Single, _
        XSiO2 As Single, XTiO2 As Single, XAl2O3 As Single, XFeO As Single, _
        XMgO As Single, XCaO As Single, XNa2O As Single, XK2O As Single)

Dim sum_mol_prop_ox As Single 'Sum of the mole proportion oxides

sum_mol_prop_ox = mol_prop_ox(SiO2wp, MWSiO2) + mol_prop_ox(TiO2wp, MWTiO2) _
            + mol_prop_ox(Al2O3wp, MWAl2O3) + mol_prop_ox(FeOwp, MWFeO) _
            + mol_prop_ox(MgOwp, MWMgO) + mol_prop_ox(CaOwp, MWCaO) _
            + mol_prop_ox(Na2Owp, MWNa2O) + mol_prop_ox(K2Owp, MWK2O)


XSiO2 = (mol_prop_ox(SiO2wp, MWSiO2) / sum_mol_prop_ox)
XTiO2 = (mol_prop_ox(TiO2wp, MWTiO2) / sum_mol_prop_ox)
XAl2O3 = (mol_prop_ox(Al2O3wp, MWAl2O3) / sum_mol_prop_ox)
XFeO = (mol_prop_ox(FeOwp, MWFeO) / sum_mol_prop_ox)
XMgO = (mol_prop_ox(MgOwp, MWMgO) / sum_mol_prop_ox)
XCaO = (mol_prop_ox(CaOwp, MWCaO) / sum_mol_prop_ox)
XNa2O = (mol_prop_ox(Na2Owp, MWNa2O) / sum_mol_prop_ox)
XK2O = (mol_prop_ox(K2Owp, MWK2O) / sum_mol_prop_ox)

End Sub
```

```
Function mol_prop_ox(wtperoxide As Single, MWoxide As Single) 'Calculates Wt%/MW

mol_prop_ox = wtperoxide / MWoxide

End Function
```

Next is *CalcXhydrous*. This is arguably the most important subroutine of the program. This is what takes anhydrous mole fraction oxide values and H2O and CO2 mole fraction values and converts those into hydrous bulk mole fraction values.

```
Sub CalcXhydrous(XSiO2anh As Single, XTiO2anh As Single, XAl2O3anh As Single, _
            XFeOanh As Single, XMgOanh As Single, XCaOanh As Single, _
            XNa2Oanh As Single, XK2Oanh As Single, XH2Oanh As Single, XCO2anh As Single, _
            XSiO2hyd As Single, XTiO2hyd As Single, XAl2O3hyd As Single, _
            XFeOhyd As Single, XMgOhyd As Single, XCaOhyd As Single, XNa2Ohyd As Single, _
            XK2Ohyd As Single)

Dim sum_mol_prop_hyd As Single 'Sum of the hydrous mole proportion oxides
Dim XH2Otot As Single, XCO2tot As Single, anhydrous_mol_frac_sum As Single

anhydrous_mol_frac_sum = XSiO2anh + XTiO2anh + XAl2O3anh + XFeOanh + XMgOanh + _
                         XCaOanh + XNa2Oanh + XK2Oanh + XH2Oanh + XCO2anh

XH2Otot = XH2Oanh + (XH2Oanh) ^ 2 + (XH2Oanh) ^ 3 + (XH2Oanh) ^ 4 + (XH2Oanh) ^ 5 / anhydrous_mol_frac_sum

XCO2tot = XCO2anh + (XCO2anh) ^ 2 + (XCO2anh) ^ 3 + (XCO2anh) ^ 4 + (XH2Oanh) ^ 5 / anhydrous_mol_frac_sum


sum_mol_prop_hyd = XSiO2anh + XTiO2anh + XAl2O3anh + XFeOanh + XMgOanh _
           + XCaOanh + XNa2Oanh + XK2Oanh + XH2Otot + XCO2tot


XSiO2hyd = (XSiO2anh / sum_mol_prop_hyd)
XTiO2hyd = (XTiO2anh / sum_mol_prop_hyd)
XAl2O3hyd = (XAl2O3anh / sum_mol_prop_hyd)
XFeOhyd = (XFeOanh / sum_mol_prop_hyd)
XMgOhyd = (XMgOanh / sum_mol_prop_hyd)
XCaOhyd = (XCaOanh / sum_mol_prop_hyd)
XNa2Ohyd = (XNa2Oanh / sum_mol_prop_hyd)
XK2Ohyd = (XK2Oanh / sum_mol_prop_hyd)


End Sub
```

NormXhydrous is a subroutine that helps to eliminate error in the hydrous bulk value calculation. This is where all error in the final output values comes in. Due to the nature of the calculation, you must normalize oxide values without normalizing H2O and CO2 values (H2O and CO2 values are measured, so those must remain constant). This normalization causes some error, which can be corrected for in all H2O and CO2 values below about 0.2 mole fraction. This is good because even 0.2 mole fraction values are much to high to be realistic. Any realistic values you put in will have negligible error.

```
Sub NormXhydrous(XSiO2er As Single, XTiO2er As Single, XAl2O3er As Single, _
            XFeOer As Single, XMgOer As Single, XCaOer As Single, _
            XNa2Oer As Single, XK2Oer As Single, XH2Oer As Single, XCO2er As Single, _
            XSiO2norm As Single, XTiO2norm As Single, XAl2O3norm As Single, _
            XFeOnorm As Single, XMgOnorm As Single, XCaOnorm As Single, _
            XNa2Onorm As Single, XK2Onorm As Single)

Dim sum_mol_prop_norm As Single 'Re-sum of hydrous mole proportion oxides

sum_mol_prop_norm = XSiO2er + XTiO2er + XAl2O3er + XFeOer + XMgOer _
           + XCaOer + XNa2Oer + XK2Oer + XH2Oer + XCO2er

XSiO2norm = (XSiO2er / sum_mol_prop_norm)
XTiO2norm = (XTiO2er / sum_mol_prop_norm)
XAl2O3norm = (XAl2O3er / sum_mol_prop_norm)
XFeOnorm = (XFeOer / sum_mol_prop_norm)
XMgOnorm = (XMgOer / sum_mol_prop_norm)
XCaOnorm = (XCaOer / sum_mol_prop_norm)
XNa2Onorm = (XNa2Oer / sum_mol_prop_norm)
XK2Onorm = (XK2Oer / sum_mol_prop_norm)

End Sub
```

Calcwtperhydrous simply converts the calculated hydrous bulk mole fraction values into hydrous weight percent values. This is useful because many scientists are more familiar with weight percent values. The fact that the program displays both mole fraction and weight percent values is highly useful. Similar to function *mol_prop_ox,* function *mole_times_MW* defines the mathematical function used in the conversion.

```
Sub Calcwtperhydrous(molSi02h As Single, molTi02h As Single, molAl203h As Single, molFe0h As Single, _
                molMg0h As Single, molCa0h As Single, molNa20h As Single, molK20h As Single, _
                molH20h As Single, molC02h As Single, _
                wtSi02h As Single, wtTi02h As Single, wtAl203h As Single, wtFe0h As Single, _
                wtMg0h As Single, wtCa0h As Single, wtNa20h As Single, wtK20h As Single, _
                wtH20h As Single, wtC02h As Single)

Dim sum_mol_times_MW As Single 'Sum of X*MW values

sum_mol_times_MW = mol_times_MW(molSi02h, MWSi02) + mol_times_MW(molTi02h, MWTi02) _
             + mol_times_MW(molAl203h, MWAl203) + mol_times_MW(molFe0h, MWFe0) _
             + mol_times_MW(molMg0h, MWMg0) + mol_times_MW(molCa0h, MWCa0) _
             + mol_times_MW(molNa20h, MWNa20) + mol_times_MW(molK20h, MWK20) _
             + mol_times_MW(molH20h, MWH20) + mol_times_MW(molC02h, MWC02)

wtSi02h = (mol_times_MW(molSi02h, MWSi02) / sum_mol_times_MW) * 100
wtTi02h = (mol_times_MW(molTi02h, MWTi02) / sum_mol_times_MW) * 100
wtAl203h = (mol_times_MW(molAl203h, MWAl203) / sum_mol_times_MW) * 100
wtFe0h = (mol_times_MW(molFe0h, MWFe0) / sum_mol_times_MW) * 100
wtMg0h = (mol_times_MW(molMg0h, MWMg0) / sum_mol_times_MW) * 100
wtCa0h = (mol_times_MW(molCa0h, MWCa0) / sum_mol_times_MW) * 100
wtNa20h = (mol_times_MW(molNa20h, MWNa20) / sum_mol_times_MW) * 100
wtK20h = (mol_times_MW(molK20h, MWK20) / sum_mol_times_MW) * 100
wtH20h = (mol_times_MW(molH20h, MWH20) / sum_mol_times_MW) * 100
wtC02h = (mol_times_MW(molC02h, MWC02) / sum_mol_times_MW) * 100

End Sub
```

```
Function mol_times_MW(moloxide As Single, MWoxide As Single) 'Calculates X*MW

mol_times_MW = moloxide * MWoxide

End Function
```

SendoutputX and *Sendoutputwtper* each send hydrous mole fraction and weight percent values to the output columns on the worksheet, respectively.

```
Sub SendoutputX(XSiO2o As Single, XTiO2o As Single, XAl2O3o As Single, XFeOo As Single, _
            XMgOo As Single, XCaOo As Single, XNa2Oo As Single, XK2Oo As Single, _
            XH2Oo As Single, XCO2o As Single)

    Dim outputRange As Object, theColumn As Integer

    Set sheet = Application.Workbooks("Conversion_tool.xls").Worksheets("Worksheet")
    Set outputRange = sheet.Range("J5")

    Range("Xhyd").Select

    theColumn = 1

    Do While Val(outputRange.Cells(theColumn, 2).Value) > 0
       theColumn = theColumn + 1
    Loop

    With outputRange
       .Cells(theColumn, 1).Value = XSiO2h
       .Offset(1, 0).Value = XTiO2h
       .Offset(2, 0).Value = XAl2O3h
       .Offset(3, 0).Value = XFeOh
       .Offset(4, 0).Value = XMgOh
       .Offset(5, 0).Value = XCaOh
       .Offset(6, 0).Value = XNa2Oh
       .Offset(7, 0).Value = XK2Oh
       .Offset(8, 0).Value = XH2Oin
       .Offset(9, 0).Value = XCO2in
    End With

End Sub
```

SaveData, Clear_OutputX, Clear_OutputWt, Clear_Input, and *Clear_SaveData* all define buttons on the excel worksheet.

Example with Real Data

For the final part of this walkthrough tutorial, I will demonstrate how this program operates with real life data. The data set I have chosen is an anhydrous analysis of a phonotephrite lava from Mt. Erebus in Antarctica. I have chosen a calculation with an H2O mole fraction of .08 and a CO2 mole fraction of .01.

Anhydrous composition:

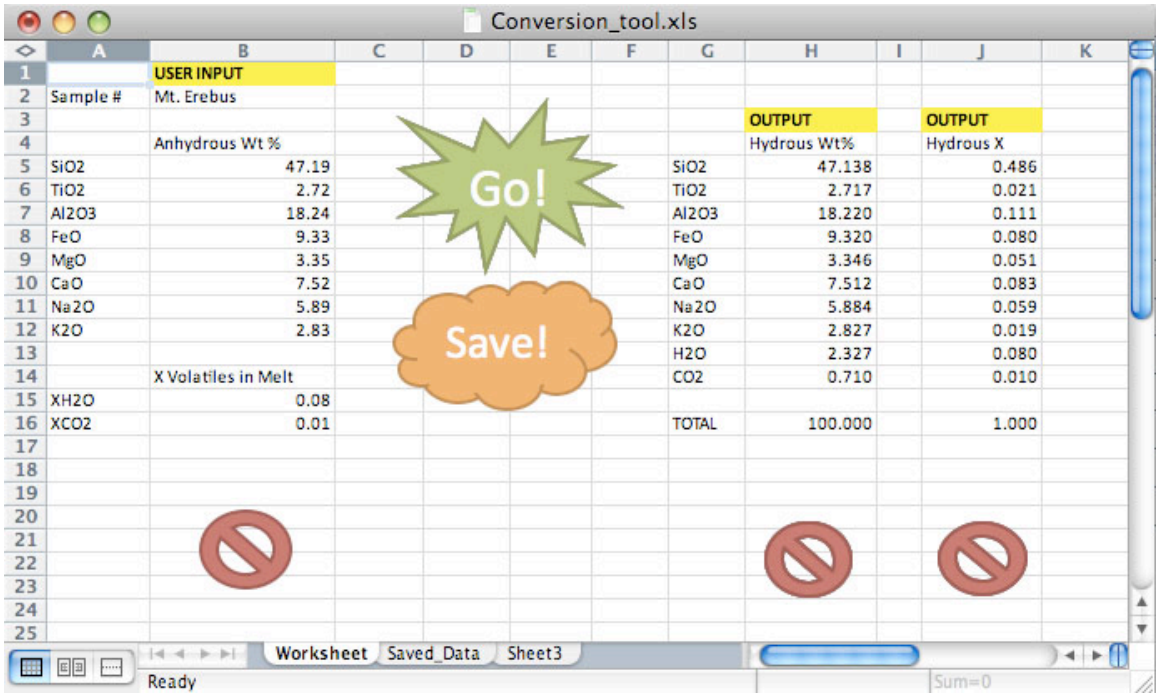
Oxide	Values in Wt. %
SiO2	47.19
TiO2	2.72
Al2O3	18.24
FeO	9.33
MgO	3.35
CaO	7.52
Na2O	5.89
K2O	2.83

Volatiles dissolved in melt:

	Mole Fraction
H2O	.08
CO2	.01

After running the program, I got the following output hydrous values:

	Hydrous Wt. %	**Hydrous X**
SiO2	47.138	0.486
TiO2	2.717	0.021
Al2O3	18.220	0.111
FeO	9.320	0.080
MgO	3.346	0.051
CaO	7.512	0.083
Na2O	5.884	0.059
K2O	2.827	0.019
H2O	2.327	0.080
CO2	0.710	0.010
TOTAL	**100.000**	**1.000**



Where doesn't this model work well?

As stated in the tutorial, the calculation works very well for all mole fraction values of volatiles below 0.2. Above 0.2, the error increases too much for the calculations to be accurate enough to use. This is okay, however, because even 0.2 is an order of magnitude higher than a typical value for mole fraction of H2O or CO2.

What do these results mean?

The results calculated in this program are very useful to scientists studying the deep Earth. Sometimes, a geologist might have a rock that is known to have contained H_2O and CO_2 when it was a melt at depth. We get this information via melt inclusion analysis, which gives us the fixed H_2O and CO_2 values in a melt at depth. However, when the rock has made it to the surface, it generally has lost all of its H_2O and CO_2. So, any bulk analyses we can take here at the surface of the oxide concentrations will be anhydrous values. In order to determine the relationship between different oxide concentrations and H_2O and CO_2 solubility in these melts, we must combine these two measurements into a hydrous concentration analysis.

In the case of the example data, Mt. Erebus phonotephrite, the anhydrous values were known from bulk analysis, but the mole fraction of H_2O and CO_2 were not known. However, it is known roughly how much H_2O and CO_2 would be present in this melt at saturation. With this, we can create a hypothetical situation to test what would happen with differing amounts of H_2O and CO_2 dissolved in this particular composition.

This program is also very useful in testing how accurately a model for H_2O and CO_2 solubility in a silicate melt will perform. When modeling this, the user inputs several model parameters based upon experimentally constrained data. These parameters will produce different models. We can run these models, determine the H_2O and CO_2 solubility, and plug in those values into this program. With the ability to save several data sets onto the "Saved Data" sheet, we can observe many models at one time and compare how well they perform relative to one another.

In future work, this program also has the ability to be upgraded into an even more complex program that could calculate H_2O and CO_2 solubility values based on particular compositional variables. This would be a much more extensive task, as it would require large amounts of experimental data and modeling tools. As it is, this program is extremely useful for deep Earth scientists (and those casually interested in hydrous oxide concentrations!), and it has the potential to be expanded into an even more advanced program.