

# Igor

## Technical Notes

---

WaveMetrics Guest

### #021: Ternary Graphs

Written by:  
(guest writer)  
Updated by:  
Hutchinson (staff)

Jozsef Urmos  
June 1990  
Larry  
July 1990

This Technical Note describes the creation of Ternary graphs with Igor.

#### **Accompanying files:**

'Ternary' -- ternary graph stationery experiment  
'SSData.awav' -- example data

---

Ternary graphs provide a useful means for exploring the relationships among three variables. Data plotted on a ternary are limited by the requirement that they be normalized so that the values for each of the three components are expressed as proportions of the whole data set. Examples of the types of data that are naturally amenable to display on a ternary are compositional analyses (chemical or physical) in which the abundances of each of the components are reported on a percentage basis.

A typical use of a ternary diagram is for classification based on three characteristic components drawn from a compositional analysis, and determining in what particular field(s) these data may cluster on the diagram. (See the example data set, illustrated on the last page, which presents a classification scheme for sandstones based on detrital mineral abundances.) More specialized uses are found in thermodynamics where ternary diagrams are used to examine the phase equilibria between the various chemical components in a system.

Statistical analysis of these kinds of constant sum, closed data systems is a difficult and messy business. Other limitations of ternaries, in particular, are discussed in the paper by Philip and Watson (1989) and papers cited therein.

## **Implementation**

A ternary graph or diagram is simply an equilateral triangle on which each vertex represents the pure end-member of each component. The side opposite each vertex represents a zero value of that particular component.

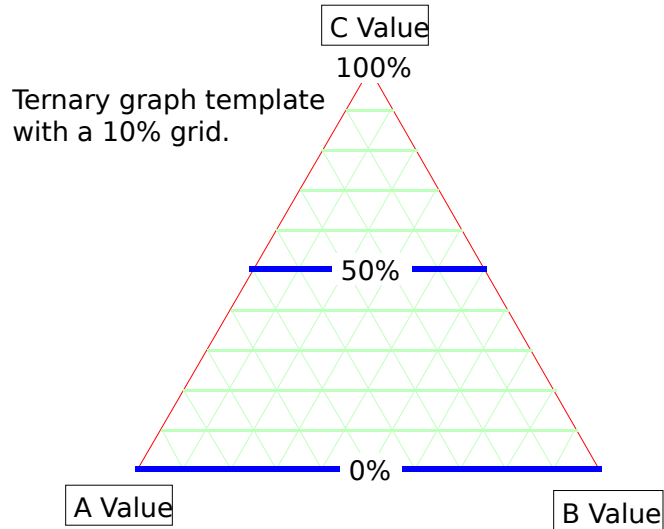
For graphing in Igor, the ternary data components (A, B, C) are transformed into Cartesian (x, y) coordinates with the following two equations:

$$x = 0.5 * C + B \quad (1)$$

$$y = C * \cos( 30^\circ ). \quad (2)$$

These equations produce "ternary" data such that the A component vertex is situated at the origin, the base of the ternary is defined by the A & B components that lie along the x-axis, and the C

vertex is located at the ternary apex. Such a ternary is illustrated below. This ternary also illustrates where the C component will have values of 0%, 50%, and 100%.



## Ternaries in Igor

The 'Ternary' experiment supplied with this TN includes three different ternary templates to which you may append data. The templates are created by the graph macros:

- Ternary — ternary boundaries
- TernTicks — ternary with tick marks at 10% or 0.1 unit intervals
- TernGrid — ternary with a 10% or 0.1 unit grid (shown above).

The `GrafTernData()` and `ConvertF2P2F()` utility macros are included to help simplify the graphing of data on a ternary, and to change the data defining the ternaries from a fractional basis to percentage or back again according to the form of your data.

.....  
| The `GrafTernData()` macro converts normalized data waves provided by you  
| the user into the proper format for plotting on a new ternary or to be

appended to an appropriate target ternary.

The following information is specified through a missing parameter dialog:

- (1) Names of the three waves to be graphed. Only the B and C component waves are required for the coordinate transform. The wave names supply the axis labels.
- (2) One of the three types of ternaries to be displayed. This is optional if appending to a ternary graph.
- (3) Names for the X and Y waves that will contain the transformed data for graphing on the ternary.
- (4) Create a new ternary or append to an existing one.

The new X and Y waves that are created will automatically overwrite any waves that already have the same name(s).

---

```

Macro GrafTernData( Awave, Bwave, Cwave, type, Xwave, Ywave, doNew )
  String      Awave
  Prompt Awave, "Select the A component wave :", popup, WaveList("*",";", "")
  String      Bwave
  Prompt Bwave, "Select the B component wave :", popup, WaveList("*",";", "")
  String      Cwave
  Prompt Cwave, "Select the C component wave :", popup, WaveList("*",";", "")
  Variable    type
  Prompt type, "Ternary graph type :", popup, "Ternary;Ticked Tern;Gridded Tern"
  String      Xwave = "x_Twave"
  Prompt Xwave, "Name for the ternary X-data wave:"
  String      Ywave = "y_Twave"
  Prompt Ywave, "Name for the ternary Y-data wave:"
  Variable    doNew
  Prompt doNew, "What to Do?", popup, "New Ternary;Append"

```

|.....

| ConvertF2P2F() is a utility macro to convert the data waves defining the  
 | ternary graphs to a percent basis or back to fractions again to accommodate  
 | the type of data you are graphing on the ternary.

Macro **ConvertF2P2F**()

The axes, grid, and ticks waves contain the data used to create the respective parts of the ternaries.

## Plotting your data:

Graphing your data on a ternary is fairly straightforward. First you must make certain to normalize your data to either one (1) or to 100%. Next, you import your data into the Ternary experiment and use the `GrafTernData()` macro to transform and plot your data on a new ternary diagram. You can also use `GrafTernData()` to transform and append data to the topmost ternary graph window; in this case you need not specify the ternary graph type or the A component wave. If you have used equations 1 and 2 to transform your data beforehand, then you need only select the graph macro for the ternary you want and append your data parametrically.

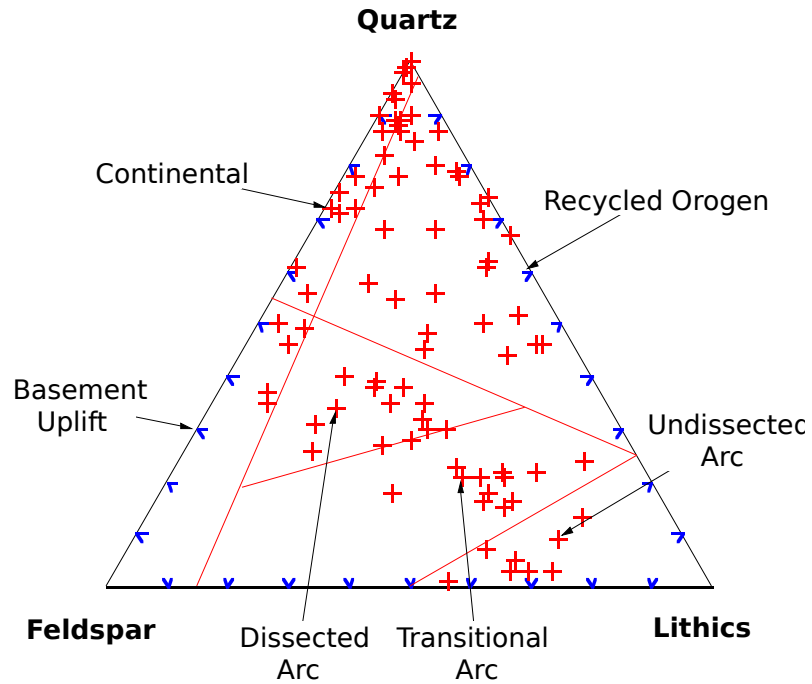
By default, it is assumed that your data are normalized to one (1). If your data are on a percentage basis, then run the `ConvertF2P2F()` macro to change the ternaries to the same scale. If you typically use percent data, then you might consider saving the Ternary experiment with percent as the default.

Alternatively, you may want to import the Ternary procedures into one of your own experiments. If you do, remember also to import the ternary data waves that you will need to create the ternaries.

If your data are not properly displayed on a ternary, verify that the data types agree (eg. percent data on a percent ternary), or that the transformed data are not negative or exceed  $x = 1$  (or 100%) and  $y = 0.866$  (or 86.6%) which indicates an error in the normalization or transformation.

Once you have plotted your data, you may freely resize the graph without distorting the ternary by changing its aspect ratio. As with any constrained graph in a page layout, it is probably best to resize the graphs manually and not use the tile feature or any other automatic resizing feature before printing.

Included with this TN are some example data obtained from the Dickinson references below. Here's how the sample data should look. The xxBdry's data were used to define the various field lines on the figure. All data are part of the text file.



### Further Reference:

- Dickinson, W.R., and Suczek, C.A. (1979) Plate tectonics and sandstone compositions. *American Association of Petroleum Geologists Bulletin*, v. 63, p. 2164-2182.
- Dickinson, W.R., and others (1983) Provenance of North American Phanerozoic sandstones in relation to tectonic setting. *Geological Society of America Bulletin*, v. 94, p. 222-235.
- Philip, G.M., and Watson, D.F. (1989) Some geometric aspects of the ternary diagram. *Journal of Geological Education*, v. 37, p 27-29.