TERNPLOT - SAS® Creation of Ternary Plots

by Leon L. Fedenczuk - Canadian Hunter Exploration Ltd.
Mark Bercov - Bercov Computer Consultants Ltd.

BEST CONTRIBUTED PAPER

TERNPLOT is a SAS application which produces triangular (ternary) plots on various graphics devices. These plots allow the user to display three coordinate system data on a two dimensional graph. Each apex of the ternary plot represents one dimension of the coordinate system. Different symbols and colors may be utilized for the display of each distinct group of input observations to aid in the interpretation of results.

Ternary plots allow the display of more information per graph than any form of crossplot. TERNPLOT increases the ability of the user to easily display and interpret information for complex systems such as rock mineralogy, multiphase flow, chemical analysis, or the output from multivariate statistical procedures.

THEORY

Data in the ternary plot is placed inside of an equilateral triangle. This type of data representation is well suited for the graphical analysis of closed systems, where the sum of the parameters for each observation is always constant. The original data (which normally represents three coordinates in an open system) is transformed into data forming a closed system by normalizing the original data values (A, B, and C) for each observation into the range 0...1 as proportions of the sum of these three data values as noted below.

AA = A / (A + B + C)
BB = B / (A + B + C)
CC = C / (A + B + C)

The transformation above constitutes the closure of the system, even though the original data represents an open system with no constraints on any of the original components or their sum. Such a transformation should not be applied to any three arbitrary components, but only to three components which, when added together, constitute the entirety of the entity under investigation. In practical terms, there are only two independent dimensions, because the transformed third dimension component is always equal to one minus the sum of the transformed components for the first two dimensions.

CALCULATION OF X, Y DATA COORDINATES

The transformed three coordinate system data is reduced to X and Y values which lie inside of an equilateral triangle. This data is then plotted using a fourth parameter as a symbol and a fifth parameter for color selection. The use of the symbol and color parameters increases the number of displayed dimensions, and is especially useful in discriminating different populations in the analyzed data.

Graphically, the data reduction from the transformed three coordinate system data ratios to the X and Y device independent coordinates, in the range 0-100, and inside the triangle is defined as the intersection of the two lines corresponding to the normalized values of the parameters A and C (see the two dotted lines on FIG. 1). The line parallel to BC divides AA1 according to the value of AA (A's relative value in the ABC closed system), and the line parallel to AB divides CC1 according to the value of CC (C's relative value). Each apex of the triangle corresponds to 100% of the corresponding parameter.

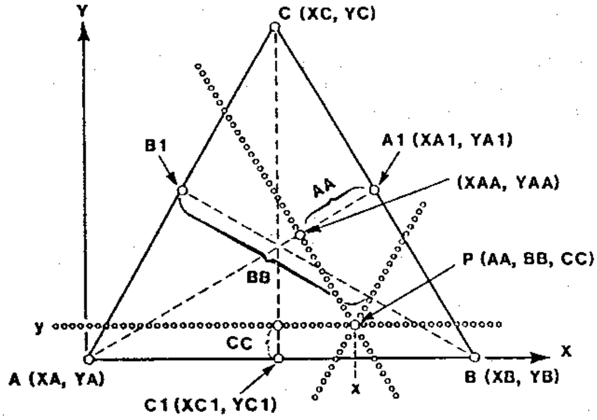


Fig. 1
Calculation of relative x,y coordinates for P(AA,BB,CC)

DETAILED ALGORITHM FOR THE CALCULATION OF X AND Y COORDINATES

```
/*Initialization and calculations for all vertices.        */
  SROOT3 = SQRT (3.0)
  XA = 0  XB = 100  XC = (XB -XA) / 2.0
  YA = 0  YB = 0    YC = XC * SROOT3
/*A1 coordinates.                                           */
  YA1 = YC / 2.0
  XA1 = YA1 * SROOT3
/*Directional coefficient of BC.                            */
  COEF_BC = (YC -YB) / (XC -XB)
/*X and Y coordinates (at point AA) for the angled          */
/*dotted line along AA1.                                    */
  XAA = XA1 * (1.0 -AA)
  YAA = XAA / SROOT3
/*Y coordinate of intersection of dotted lines              */
/*(and equation of horizontal line corresponding            */
/*to CC value).                                             */
  Y = YC * CC                              ...............(1)
/*Equation of line parallel to BC and passing               */
/*through AA.                                               */
  Y1 = COEF_BC * (X -XAA) + YAA            ...............(2)
/*The X coordinate of the intersection of the dotted        */
/*lines obtained by solving equations (1) and (2) above     */
  X = (Y-YAA)/COEF_BC+ XAA
                                           ................(3)
```

OUTLINE OF TERNPLOT

The present version of TERNPLOT has been written in the form of a typical SAS/AF® application utilizing macros to do the actual processing. The user supplies the names of the appropriate input data set variables as parameters. The program requires the names of three numeric variables (one for each dimension), an alphabetic symbol variable, and an alphabetic color variable. The user is also allowed to specify appropriate title and label information, and to choose whether or not grid lines at 10% intervals are to appear on the graph. TERNPLOT has been developed to be applied as it is or to be modified relatively easy to fit specific tasks.

SAS PROCESSING

The implemented SAS processing procedures are relatively simple and straightforward, as outlined in Figure 2. The user submits a program TERNTNGL.SAS to create a permanent SAS dataset (TERNTNGL) which contains the outline of the equilateral triangle, and submits a program (TERNGRID.SAS) to create a permanent SAS dataset (TERNGRID) containing the grid lines and grid labels. Both of these programs must be executed once only, because their output is completely independent of any data to be plotted.

At the start of each plotting session, the user is presented with an AF Screen which allows the choice of the desired graphics device (from a list which currently includes a VGA terminal, an EGA terminal, a HP Laserjet printer, a HP PaintJet plotter, a HP 7550 A size plotter, and a HP Draftmaster II plotter), and the graph orientation (Landscape or Portrait). The user then is prompted with an AF Screen which allows the specification of the input dataset name, variables to be plotted, plot title, vertex annotation, and grid specification. Once these choices are made, the appropriate SAS macro programs (which perform all of the real work) are submitted.

The majority of the SAS code executed is device independent (which has been achieved by isolating most of the calculations from graphics device characteristics). The core of the system creates and stores data and objects such as labels in the form of SAS annotate data sets. This is done in a 2D X/Y coordinate system and defined as screen/device percentages (so as to be unaffected by Title and /or Footnote scaling) in the range 0-100 as described in the algorithm above. The SAS annotate data sets created are combined and scaled appropriately for the selected graphics device, creating a final annotate data set. It should be noted that the above step is absolutely necessary.

Even though SAS supports device independent graphics output, it does not maintain aspect ratios on different devices. Consequently, if the device specific scaling described above is not performed, the final output is unlikely to be an equilateral triangle. A graph stored in a temporary graphics catalog is created from the final annotate dataset utilizing PROC GSLIDE (which also adds the appropriate titles and footnotes to the graph). The process is concluded by replaying the graph on the specified device. This procedure of creating a temporary graph, and then replaying it on the selected device allows the use of GREPLAY color maps which translate the original colors specified into the corresponding colors for the final graphics device. A brief description of each of the SAS Macro programs in the TERNPLOT system appears below:

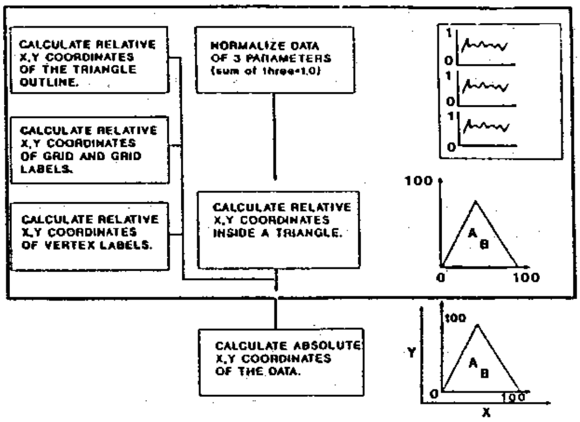
M_TPGDEF - defines selected graphics device GOPTIONS and scaling factors

M_TPPLDT - reads, checks, and normalizes input data
- calculates relative X, Y coordinates
- creates temporary annotate data set TERNDATA

M_TPVERT - outputs vertex labels
- creates temporary annotate data set TERNVERT

M_TPSCAL - reads all permanent and temporary annotate data sets
- scales data for specified graphics device
- creates temporary annotate data set TERNANNO

M_TPPLOT - creates graph in temporary graphics catalog
- replays graph on specified graphics device (with appropriate color mapping)
- performs clean up



TERNPLOT SYSTEM DIAGRAM

Fig. 2

EXCEPTION PROCESSING

Missing values are replaced with zeros and the routine performs the check to determine whether the sum of all of the numeric variables is equal to 0. In a such case, the input observation is deleted.

APPLICATIONS

The powerful capabilities of the ternary plot have been recognized and used in variety of applications in mineralogy, chemistry and statistical analysis where simple graphical methods are inadequate for the representation of data sets (see references). Each vertex of the ternary plot may represents a single parameter or a combination of parameters.

The most common applications are:

1. Representation of oil-gas-water equilibrium saturations during flow through the rock, thereby showing the region of most favourable geological conditions.
2. Presentation of selected anion/cation combinations of formation water analysis. This is applied in oil exploration by relating the chemical water types to the geology and flow patterns.
3. Presentation of fundamental rock classification in terms of Quartz-Feldspar-Rock Fragments.
4. Creation of variation diagrams displaying covariance between parameters.

5. Creation of graphical output from dimension reducing procedure Principal Component Analysis (available in SAS).

6. Creation of graphical output from Q-type factor analysis commonly used in data analysis.

7. Discrimination of data based on natural blobs (clusters) observed in the triangular plots of raw or dimension reduced data.

8. Provision of a new procedure to aid in the analysis of any 3 component closed system.

CONCLUSIONS

An efficient SAS application to create ternary plots has been described. It demonstrates an effective methodology in using SAS annotate procedure and macro facilities. This paper demonstrates how these tools can be incorporated in graphical algorithms to improve statistical and graphical data analysis.

REFERENCES

1. B. Hitchon, Graphical and Statistical treatment of Standard Formation Water Analysis, Proceedings of First Canadian/American Conference on Hydrology Practical Applications of Ground Water Geochemistry. Banff, Alberta, June 1984.

2. P. S. Mustard, J. M. Richardson, A Lotus 1-2-3Template for triangle plots, GEOBYTE, April 1990.

3. P. A. Dickey, Patterns of Chemical Composition In Deep Subsurface Waters, Geological Notes, Bulletin of the American Association of Petroleum Geologists, Vol.50, Part II, July-Dec. 1966.

4. C. G. Topley, A.D. M. Burwell, Trigplot an Interactive Program in Basic for Plotting Triangular Diagrams, Computers & Geosciences, Vol.10, No.2-3, 1984.

5. T. V. Subba Rao, D. Rameshway Rao, Recplot, Triplot - Generalized Graphics Programs for plotting geochemical variation diagrams, Geophysical Research Bulletin, Vol.26, No.1, March 1988.

If you have any questions or comments, or desire further information, the authors may be contacted at

Canadian Hunter Exploration Ltd.
435 4th Ave S.W.
CALGARY, ALBERTA
T2P 3A8, CANADA

APPENDIX - source listings

```
/* This is THE AUTOEXEC.SAS file for TERNPLOT.                              */

LIBNAME SASDATA  'SASDATA';              /* Allocate Permanent Data         */
                                         /*  Database.                      */
                                         /*                                 */
LIBNAME SASSCR   'SASSCR';               /* Allocate Permanent Screen       */
                                         /*  Database.                      */
                                         /*                                 */
%INCLUDE '\SAS\SASMACRO\ANNOMAC.SAS';/* Compile SAS Annotate Macros.        */
%INCLUDE 'SASMACS\M_TPGDEF.SAS';         /* Compile                         */
%INCLUDE 'SASMACS\M_TPPLDT.SAS';         /*  Ternplot                       */
%INCLUDE 'SASMACS\M_TPPLOT.SAS';        ,/*  Macro                          */
%INCLUDE 'SASMACS\M_TPSCAL.SAS';         /*  Programs.                      */
%INCLUDE 'SASMACS\M_TPVERT.SAS';         /*                                 */
                                         /*                                 */
FILENAME HPLJ300  'HPLJ300.LST';         /* Allocate HP Laserjet Plotter    */
                                         /*  Output Graphics Stream File.   */
                                         /*                                 */
FILENAME HPPAINT  'HPPAINT.LST';         /* Allocate HP Paintjet Plotter    */
                                         /*  Output Graphics Stream File.   */
                                         /*                                 */
FILENAME HP7550   'HP7550.LST';          /* Allocate HP 7550 Plotter        */
                                         /*  Output Graphics Stream File.   */
                                         /*                                 */
FILENAME HPDRAFT  'HPDRAFT.LST';         /* Allocate HP Draftmaster II      */
                                         /*  Plotter Output Graphics        */
                                         /*  Stream File.                   */
```

```
/*    TERNTNGL.SAS                                                          */
/*                                                                          */
/****************************************************************************/
/*                                                                          */
/*    This SAS Program creates a SAS Annotate Dataset which                 */
/*        contains the appropriate annotation output for the                */
/*        triangle outline for the Ternary Plotting System.                 */
/*                                                                          */
/*    Programmed by Leon Fedenczuk and Mark Bercov on 91-01-24              */
/*        for Canadian Hunter Exploration Ltd.                              */
/*                                                                          */
/****************************************************************************/
/*                                                                          */

    /*                                                                      */
    /*    Create the SAS Annotate Dataset which contains the                */
    /*       appropriate annotation output for the triangle                 */
    /*       outline.                                                       */
    /*                                                                      */

    DATA SASDATA.TERNTNGL
         (LABEL = 'Ternplot Triangle Outline Annotation');

    %DCLANNO;                                  /* Define length and data    */
                                               /*  type specifications      */
                                               /*  for annotate variables.  */

    ATTRIB TRIANHGT LABEL    =
                     'Relative Height of Triangle (%)          '
                    LENGTH   = 8
                    INFORMAT = 7.4
                    FORMAT   = 7.4;

    DROP TRIANHGT;

    TRIANHGT = (SQRT (0.75) * 100.0);  /* Calculate relative height  */
                                       /*  of triangle.              */

    %SYSTEM (3, 3, 3)                   /* Define all coordinates    */
                                        /*  as screen percentages.   */

    %LINE  (  0.0,   0.0,               /* Draw                      */
            100.0,   0.0,               /*  triangle                 */
             CYAN,     1,    5)         /*  outline.                 */
    %LINE  (100.0,   0.0,               /*                           */
             50.0, TRIANHGT,            /*                           */
             CYAN,     1,    5)         /*                           */
    %LINE  ( 50.0, TRIANHGT,            /*                           */
              0.0,   0.0,               /*                           */
             CYAN,     1,    5)         /*                           */

    RUN;
```

```
/*    TERNGRID.SAS                                                          */
/*                                                                          */
/**************************************************************************/
/*                                                                          */
/*    This SAS Program creates a SAS Annotate Dataset which                */
/*        contains the appropriate annotation output for the               */
/*        triangle grid for the Ternary Plotting System.                   */
/*                                                                          */
/*    Programmed by Leon Fedenczuk and Mark Bercov on 91-01-24             */
/*        for Canadian Hunter Exploration Ltd.                             */
/*                                                                          */
/**************************************************************************/
/*                                                                          */

    /*                                                                      */
    /*    Create the SAS Annotate Dataset which contains the               */
    /*        appropriate annotation output for the triangle               */
    /*        grid.                                                         */
    /*                                                                      */

DATA SASDATA.TERNGRID
     (LABEL = 'Ternplot Triangle Grid Annotation');

%DCLANNO;                                 /* Define length and data     */
                                          /*  type specifications       */
                                          /*  for annotate variables.   */

ATTRIB TEXT      LABEL    = 'Annotation Text'
                 LENGTH   = $ 14
                 INFORMAT = $14.
                 FORMAT   = $14.;

ATTRIB INDEX     LABEL    =
                   'Array Index
                 LENGTH   = 4
                 INFORMAT = 2.
                 FORMAT   = 2.;

ATTRIB TRIANHGT LABEL    =
                   'Relative Height of Triangle (%)
                 LENGTH   = 8
                 INFORMAT = 7.4
                 FORMAT   = 7.4;

ARRAY GABX      (*) GABX1    - GABX9;/* Declare array of               */
ARRAY GABY      (*) GABY1    - GABY9;/*  grid line end-point           */
ARRAY GACX      (*) GACX1    - GACX9;/*  X and Y coordinates.          */
ARRAY GACY      (*) GACY1    - GACY9;/*                                */
ARRAY GCBX      (*) GCBX1    - GCBX9;/*                                */
ARRAY GCBY      (*) GCBY1    - GCBY9;/*                                */

ARRAY ANOTA     (*) $2                /* Declare array of grid        */
                    ANOT1    - ANOT9;/*  line label values.            */

ARRAY ANOAX     (*) ANOAX1   - ANOAX9;/* Declare array of grid         */
ARRAY ANOAY     (*) ANOAY1   - ANOAY9;/*  line label value X           */
ARRAY ANOBX     (*) ANOBX1   - ANOBX9;/*  and Y coordinates.           */
ARRAY ANOBY     (*) ANOBY1   - ANOBY9;/*                               */
ARRAY ANOCX     (*) ANOCX1   - ANOCX9;/*                               */
ARRAY ANOCY     (*) ANOCY1   - ANOCY9;/*                               */

DROP INDEX
     TRIANHGT
     GABX1    - GABX9
     GABY1    - GABY9
     GACX1    - GACX9
     GACY1    - GACY9
     GCBX1    - GCBX9
     GCBY1    - GCBY9
     ANOT1    - ANOT9
     ANOAX1   - ANOAX9
     ANOAY1   - ANOAY9
     ANOBX1   - ANOBX9
     ANOBY1   - ANOBY9
     ANOCX1   - ANOCX9
     ANOCY1   - ANOCY9;

TEXT     = '              ';


TRIANHGT = (SQRT (0.75) * 100.0);   /* Calculate relative height  */
                                    /*  of triangle.               */

DO INDEX    = 1 TO 9 BY 1;

    /*                                                                  */
    /*    Calculate the grid line end-point X and Y                    */
    /*        coordinates.                                             */
    /*                                                                  */

    GABX    (INDEX) = (INDEX    * 10.0);
    GABY    (INDEX) = 0;
    GACX    (INDEX) = (INDEX    * 5.0);
    GACY    (INDEX) = (TRIANHGT * INDEX    * 0.1);
    GCBX    (INDEX) = (50.0 + (INDEX    * 5.0) );
    GCBY    (INDEX) = (TRIANHGT * (10 - INDEX)    * 0.1);

    /*                                                                  */
    /*    Calculate the grid line label values.                        */
    /*                                                                  */

    ANOTA    (INDEX) = PUT ( (INDEX    * 10), 2.);

    /*                                                                  */
    /*    Calculate the grid line label value X and Y                  */
    /*        coordinates.                                             */
    /*                                                                  */

    ANOAX    (INDEX) = (75.0 - (INDEX    * 7.5) );
    ANOAY    (INDEX) = (  (TRIANHGT - (TRIANHGT * INDEX    * 0.1) )
                       * 0.5);
    ANOBX    (INDEX) = (25.0 + (INDEX    * 7.5) );
    ANOBY    (INDEX) = ANOAY    (INDEX);
    ANOCX    (INDEX) = 50.0;
    ANOCY    (INDEX) = (TRIANHGT * INDEX    * 0.1);

END;

%SYSTEM (3, 3, 3)                   /* Define all coordinates     */
                                    /*  as screen percentages.    */

/*                                                                      */
/*    Draw and label the grid lines parallel to the line AB.           */
/*                                                                      */

DO INDEX    = 1 TO 9 BY 1;

    %LINE    (GCBX    (INDEX),          GCBY    (INDEX),
              GACX    (10 - INDEX),     GACY    (10 - INDEX),
              BLUE,                     33,
              3)

    %LABEL   (ANOCX (INDEX),            ANOCY (INDEX),
              ANOTA    (INDEX),         BLUE,
              0,                        0,
              0.9,                      DUPLEX,
              8)

END;

/*                                                                      */
/*    Draw and label the grid lines parallel to the line AC.           */
/*                                                                      */

DO INDEX    = 1 TO 9 BY 1;

    %LINE    (GABX    (INDEX),          GABY    (INDEX),
              GCBX    (INDEX),          GCBY    (INDEX),
              BLUE,                     33,
              3)

    %LABEL   (ANOBX (INDEX),            ANOBY (INDEX),
              ANOTA    (INDEX),         BLUE,
              -120,                     0,
              0.9,                      DUPLEX,
              8)

END;

/*                                                                      */
/*    Draw and label the grid lines parallel to the line BC.           */
/*                                                                      */

DO INDEX    = 1 TO 9 BY 1;

    %LINE    (GABX    (INDEX),          GABY    (INDEX),
              GACX    (INDEX),          GACY    (INDEX),
              BLUE,                     33,
              3)

    %LABEL   (ANOAX (INDEX),            ANOAY (INDEX),
              ANOTA    (INDEX),         BLUE,
              120,                      0,
              0.9,                      DUPLEX,
              8)

END;

RUN;
```

```
/*    M_TPGDEF.SAS                                                    */
/*                                                                    */
/**********************************************************************/
/*                                                                    */
/*   The SAS MACRO Program M_TPGDEF sets the graphics options,        */
/*       scaling values, and color map to the appropriate             */
/*       values for the specified graphics device for the             */
/*       Ternary Plotting System.                                     */
/*                                                                    */
/*   Programmed by Leon Federczuk and Mark Bercov on 91-02-05         */
/*       for Canadian Hunter Exploration Ltd.                         */
/*                                                                    */
/**********************************************************************/
/*                                                                    */

%MACRO M_TPGDEF;                        /*                            */

   %GLOBAL M_COLMAP                     /* Define Global              */
           M_HVCHAR                     /*  Macro variables.          */
           M_XBASE                      /*                            */
           M_XSCAL                      /*                            */
           M_YBASE                      /*                            */
           M_YSCAL;                     /*                            */

   %LOCAL  M_GDVNAM;                    /* Define Local               */
                                        /*  Macro variable.           */

   /*                                                                 */
   /*   Set all of the common graphics options to the                 */
   /*       appropriate values.                                       */
   /*                                                                 */

   GOPTIONS RESET     = GOPTIONS
            BORDER
            NOCELL
            NODISPLAY
            GOUTMODE  = APPEND
            GSFMODE   = OFF;

   /*                                                                 */
   /*   Set the remaining graphics options, the scaling               */
   /*       values, and the color map to the appropriate              */
   /*       values for the specified graphics device.                 */
   /*                                                                 */

   %LET M_GDVNAM = %BQUOTE (%UPCASE(&M_GDEVNM));

   %IF (&M_GDVNAM EQ  )
      %THEN %LET M_GDVNAM = VGAL;

   %IF (&M_GDVNAM EQ VGAL)              /* Check for landscape        */
        %THEN %DO;                      /*  oriented VGA terminal.    */

                GOPTIONS DEVICE    = VGA
                         CBACK     = GOLD
                         ROTATE    = LANDSCAPE;

                %LET M_XBASE  = 24.260;
                %LET M_XSCAL  =  0.5148;
                %LET M_YBASE  = 14.750;
                %LET M_YSCAL  =  0.7050;
                %LET M_HVCHAR =  2.600;

                %LET M_COLMAP = BASEVGA;

           %END;                        /* End landscape oriented     */
                                        /*  VGA terminal.             */

     %ELSE %IF (&M_GDVNAM EQ VGAP)      /* Check for portrait         */
        %THEN %DO;                      /*  oriented VGA terminal.    */

                GOPTIONS DEVICE    = VGA
                         CBACK     = GOLD
                         ROTATE    = PORTRAIT;

                %LET M_XBASE  = 14.750;
                %LET M_XSCAL  =  0.7050;
                %LET M_YBASE  = 24.260;
                %LET M_YSCAL  =  0.5148;

                %LET M_HVCHAR =  2.000;

                %LET M_COLMAP = BASEVGA;

           %END;                        /* End portrait oriented      */
                                        /*  EGA terminal.             */

     %ELSE %IF (&M_GDVNAM EQ EGAL)      /* Check for landscape        */
        %THEN %DO;                      /*  oriented EGA terminal.    */

                GOPTIONS DEVICE    = EGAL
                         CBACK     = GOLD
                         ROTATE    = LANDSCAPE;

                %LET M_XBASE  = 24.400;
                %LET M_XSCAL  =  0.5120;
                %LET M_YBASE  = 14.485;
                %LET M_YSCAL  =  0.7103;
                %LET M_HVCHAR =  2.600;

                %LET M_COLMAP = BASEEGA;

           %END;                        /* End landscape oriented     */
                                        /*  EGA terminal.             */

     %ELSE %IF (&M_GDVNAM EQ EGAP)      /* Check for portrait         */
        %THEN %DO;                      /*  oriented EGA terminal.    */

                GOPTIONS DEVICE    = EGAL
                         CBACK     = GOLD
                         ROTATE    = PORTRAIT;

                %LET M_XBASE  = 14.485;
                %LET M_XSCAL  =  0.7103;
                %LET M_YBASE  = 24.400;
                %LET M_YSCAL  =  0.5120;
                %LET M_HVCHAR =  2.000;

                %LET M_COLMAP = BASEEGA;

           %END;                        /* End portrait oriented      */
                                        /*  EGA terminal.             */

     %ELSE %IF (&M_GDVNAM EQ            /* Check for landscape        */
                HPLJ300L)               /*  oriented HP Laserjet      */
        %THEN %DO;                      /*  II plotter.               */

                GOPTIONS DEVICE    = HPLJ300
                         FCACHE    = 0
                         GSFNAME   = HPLJ300
                         ROTATE    = LANDSCAPE;

                %LET M_XBASE  = 20.000;
                %LET M_XSCAL  =  0.6000;
                %LET M_YBASE  = 12.050;
                %LET M_YSCAL  =  0.7590;
                %LET M_HVCHAR =  2.000;

                %LET M_COLMAP = BASEHPLJ;

           %END;                        /* End landscape oriented     */
                                        /*  HP Laserjet II plotter.   */

     %ELSE %IF (&M_GDVNAM EQ            /* Check for portrait         */
                HPLJ300P)               /*  oriented HP Laserjet      */
        %THEN %DO;                      /*  II plotter.               */

                GOPTIONS DEVICE    = HPLJ300
                         FCACHE    = 0
                         GSFNAME   = HPLJ300
                         ROTATE    = PORTRAIT;

                %LET M_XBASE  = 12.050;
                %LET M_XSCAL  =  0.7590;
                %LET M_YBASE  = 20.000;
                %LET M_YSCAL  =  0.6000;
                %LET M_HVCHAR =  1.600;

                %LET M_COLMAP = BASEHPLJ;

           %END;                        /* End portrait oriented      */
                                        /*  HP Laserjet II plotter.   */

     %ELSE %IF (&M_GDVNAM EQ            /* Check for landscape        */
                HPPAINTL)               /*  oriented (CHEL Version)   */
        %THEN %DO;                      /*  HP Paintjet plotter.      */

                GOPTIONS DEVICE    = HPPAINT
                         FCACHE    = 0
                         GSFNAME   = HPPAINT
                         ROTATE    = LANDSCAPE;

                %LET M_XBASE  = 19.925;
                %LET M_XSCAL  =  0.6015;
                %LET M_YBASE  = 12.500;
                %LET M_YSCAL  =  0.7500;
                %LET M_HVCHAR =  2.000;

                %LET M_COLMAP = BASEHPPJ;

           %END;                        /* End landscape oriented     */
                                        /*  (CHEL Version) HP         */
                                        /*  Paintjet plotter.         */

     %ELSE %IF (&M_GDVNAM EQ            /* Check for portrait         */
                HPPAINTP)               /*  oriented (CHEL Version)   */
        %THEN %DO;                      /*  HP Paintjet plotter.      */

                GOPTIONS DEVICE    = HPPAINT
                         FCACHE    = 0
                         GSFNAME   = HPPAINT
                         ROTATE    = PORTRAIT;

                %LET M_XBASE  = 12.500;
                %LET M_XSCAL  =  0.7500;
                %LET M_YBASE  = 19.925;
                %LET M_YSCAL  =  0.6015;
                %LET M_HVCHAR =  1.600;
```

```
            %LET M_COLMAP = BASEHPPJ;

        %END;                       /* End portrait oriented     */
                                    /*  (CHEL Version) HP        */
                                    /*  Paintjet plotter.        */

%ELSE %IF (&M_GDVNAM EQ             /* Check for landscape       */
           HP7550L)                 /*  oriented HP 7550         */
    %THEN %DO;                      /*  (A Size) plotter.        */

            GOPTIONS DEVICE   = HP7550A
                     AUTOFEED
                     GSFNAME  = HP7550
                     NOPROMPT
                     ROTATE   = LANDSCAPE;

            %LET M_XBASE  = 22.065;
            %LET M_XSCAL  =  0.5587;
            %LET M_YBASE  = 11.195;
            %LET M_YSCAL  =  0.7761;
            %LET M_HVCHAR =  2.100;

            %LET M_COLMAP = BASEHP75;

        %END;                       /* End landscape oriented    */
                                    /*  HP 7550 (A Size)         */
                                    /*  plotter.                 */

%ELSE %IF (&M_GDVNAM EQ             /* Check for portrait        */
           HP7550P)                 /*  oriented HP 7550         */
    %THEN %DO;                      /*  (A Size) plotter.        */

            GOPTIONS DEVICE   = HP7550A
                     AUTOFEED
                     GSFNAME  = HP7550
                     NOPROMPT
                     ROTATE   = PORTRAIT;

            %LET M_XBASE  = 11.195;


            %LET M_XSCAL  =  0.7761;
            %LET M_YBASE  = 22.065;
            %LET M_YSCAL  =  0.5587;
            %LET M_HVCHAR =  1.550;

            %LET M_COLMAP = BASEHP75;

        %END;                       /* End portrait oriented     */
                                    /*  HP 7550 (A Size)         */
                                    /*  plotter.                 */

%ELSE %IF (&M_GDVNAM EQ             /* Check for landscape       */
           HPDRAFTL)                /*  oriented (CHEL Version)  */
    %THEN %DO;                      /*  HP Draftmaster II        */
                                    /*  plotter.                 */

            GOPTIONS DEVICE   = HPDRAFT
                     GSFNAME  = HPDRAFT
                     ROTATE   = LANDSCAPE;

            %LET M_XBASE  = 21.280;
            %LET M_XSCAL  =  0.5744;
            %LET M_YBASE  = 13.235;
            %LET M_YSCAL  =  0.7353;
            %LET M_HVCHAR =  2.000;

            %LET M_COLMAP = BASEHPDR;

        %END;                       /* End landscape oriented    */
                                    /*  (CHEL Version) HP        */
                                    /*  Draftmaster II plotter.  */

%ELSE %IF (&M_GDVNAM EQ             /* Check for portrait        */
           HPDRAFTP)                /*  oriented (CHEL Version)  */
    %THEN %DO;                      /*  HP Draftmaster II        */
                                    /*  plotter.                 */

            GOPTIONS DEVICE   = HPDRAFT
                     GSFNAME  = HPDRAFT
                     ROTATE   = PORTRAIT;

            %LET M_XBASE  = 13.235;
            %LET M_XSCAL  =  0.7353;
            %LET M_YBASE  = 21.280;
            %LET M_YSCAL  =  0.5744;
            %LET M_HVCHAR =  1.600;

            %LET M_COLMAP = BASEHPDR;

        %END;                       /* End portrait oriented     */
                                    /*  (CHEL Version) HP        */
                                    /*  Draftmaster II plotter.  */

%ELSE   %DO;                        /* Unknown graphics device - */
                                    /*  define as landscape      */
                                    /*  oriented VGA terminal.   */

            GOPTIONS DEVICE   = VGA
                     CBACK    = GOLD
                     ROTATE   = LANDSCAPE;

            %LET M_XBASE  = 24.260;
            %LET M_XSCAL  =  0.5148;
            %LET M_YBASE  = 14.750;
            %LET M_YSCAL  =  0.7050;
            %LET M_HVCHAR =  2.600;

            %LET M_COLMAP = BASEVGA;

        %END;                       /* End unknown graphics      */
                                    /*  device.                  */

%MEND M_TPGDEF;


/*     M_TPPLOT.SAS                                                   */
/*                                                                    */
/**********************************************************************/
/*                                                                    */
/*     The SAS MACRO Program M_TPPLOT reads the input data            */
/*          from the specified SAS dataset, and creates a SAS         */
/*          Annotate Dataset which contains the appropriate           */
/*          annotation output for each data point to be               */
/*          plotted for the Ternary Plotting System.                  */
/*                                                                    */
/*     Programmed by Leon Fedenczuk and Mark Bercov on 91-02-06       */
/*          for Canadian Hunter Exploration Ltd.                      */
/*                                                                    */
/**********************************************************************/
/*                                                                    */

%MACRO M_TPPLOT;                        /*                            */

    /*                                                                */
    /*     Read the input data from the specified SAS dataset,        */
    /*          and create the SAS Annotate Dataset which             */
    /*          contains the appropriate annotation output            */
    /*          for each data point to be plotted.                    */
    /*                                                                */

    DATA TERNDATA
         (KEEP  = ANGLE    COLOR    FUNCTION HSYS     POSITION
                  ROTATE   SIZE     STYLE    TEXT     WHEN
                  X        XSYS     Y        YSYS
          LABEL = 'Plot Data Annotate Dataset');

    %DCLANNO;                           /* Define length and data     */
                                        /*  type specifications       */
                                        /*  for annotate variables.   */

    ATTRIB TEXT     LABEL    = 'Annotation Text'
                    LENGTH   = $ 14
                    INFORMAT = $14.
                    FORMAT   = $14.;

    ATTRIB HEIGHT   LABEL    = 'Annotation Height (%)'
                    LENGTH   = 4
                    INFORMAT = 6.2
                    FORMAT   = 6.2;

    RETAIN HEIGHT   &M_HVCHAR
           XA       0
           YA       0
           XB       100
           YB       0
           SROOT3   1.7320508076
           XC       50
           YC       86.60254038
           YA1      43.301270189
           XA1      75
           COEF_BC  -1.7320508076;

    TEXT     = '              ';

    %SYSTEM (3, 3, 3)                   /* Define all coordinates     */
                                        /*  as screen percentages.    */

    IF (_N_     = 1)
       THEN DO;

                                        /* Calculate height of data   */
                                        /*  point annotation.         */
              HEIGHT    = (&M_HVCHAR * 0.60);

                                        /* Calculate constants        */
                                        /*  defined in paper.         */
              XA        = 0;
              YA        = 0;
              XB        = 100;
              YB        = 0;
              SROOT3    = SQRT (3.0);
              XC        = (XB        - XA       ) * 0.5;
              YC        = (XC        * SROOT3);
              YA1       = (YC        * 0.5);
              XA1       = (YA1       * SROOT3);
              COEF_BC   = (  (YC        - YB       )
                           / (XC        - XB       ) );

         END;
```

```
    SET &M_INDSN
        (KEEP = &M_XVAR   &M_YVAR   &M_ZVAR   &M_SYMB   &M_COLOR);

    /* Normalize data - the sum of M_XVAR, M_YVAR, and M_ZVAR = 1.0 */
    IF (&M_XVAR   = .)
        THEN &M_XVAR   = 0.0;
    IF (&M_YVAR   = .)
        THEN &M_YVAR   = 0.0;
    IF (&M_ZVAR   = .)
        THEN &M_ZVAR   = 0.0;
    SUMVARS  = (&M_XVAR   + &M_YVAR   + &M_ZVAR);
    IF (SUMVARS  = 0.0)
        THEN DELETE;
    &M_XVAR   = &M_XVAR   / SUMVARS;
    &M_YVAR   = &M_YVAR   / SUMVARS;
    &M_ZVAR   = &M_ZVAR   / SUMVARS;
    /* Implementation of the algorithm from the paper. */
    XAA=XA1*(1.0-&M_XVAR);
    YAA=XAA/SROOT3;
    Y=YC*&M_ZVAR;
    X=( (Y-YAA)/COEF_BC)+XAA;

    COLOR    = &M_COLOR;

    %LABEL  (       X,          Y,          /* Draw                       */
            &M_SYMB,           *,          /*  centered                  */
                  0,          0,          /*  data                      */
             HEIGHT,      SWISS,          /*  point.                    */
                  5)                      /*                            */

    RUN;

%MEND M_TPPLOT;
-

/*    M_TPPLOT.SAS                                                       */
/*                                                                       */
/*************************************************************************/
/*                                                                       */
/*    The SAS MACRO Program M_TPPLOT creates and stores the              */
/*        final Ternary Plot in a temporary graphics catalog,           */
/*        and outputs the final Ternary Plot on the specified           */
/*        graphics device for the Ternary Plotting System.              */
/*                                                                       */
/*    Programmed by Leon Fedenczuk and Mark Bercov on 91-02-05          */
/*        for Canadian Hunter Exploration Ltd.                          */
/*                                                                       */
/*************************************************************************/
/*                                                                       */

%MACRO M_TPPLOT;                          /*                            */

    %LOCAL  M_GDEVIC                      /* Define Local               */
            M_GDEVLN                      /*  Macro variables.          */
            M_GDVNAM;                     /*                            */

    /*                                                                   */
    /*    Create and store the final Ternary Plot in a                  */
    /*        temporary graphics catalog.                               */
    /*                                                                   */

    TITLE1 C = BLACK  F = SWISS  H = 4.0 PCT J = CENTER
        "&M_TITLE";

    FOOTNOTE1 C = BLACK  F = SWISS  H = 1.0 PCT  J = RIGHT
              "Created &SYSDATE - &SYSTIME ";

    PROC GSLIDE ANNOTATE    = TERNANNO
                GOUT        = TERNPLOT
                NAME        = 'TERNPLOT'
                DESCRIPTION = 'Ternary Plot';

    RUN;

    /*                                                                   */
    /*    Set the graphics options to the appropriate values            */
    /*        to allow output on the specified graphics device.         */
    /*                                                                   */

    GOPTIONS DISPLAY;

    %LET M_GDVNAM = %BQUOTE (%UPCASE(&M_GDEVNM));

    %LET M_GDEVLN = %LENGTH (&M_GDVNAM);

    %IF (&M_GDEVLN < 2)
        %THEN %LET M_GDEVIC = VGA;
        %ELSE %LET M_GDEVIC = %BQUOTE (%SUBSTR(&M_GDVNAM, 1,
                                        (&M_GDEVLN - 1)));

    %IF (   (&M_GDEVIC EQ HPLJ300)        /* Check for plotter          */
         OR (&M_GDEVIC EQ HPPAINT)        /*  specified as graphics     */
         OR (&M_GDEVIC EQ HP7550 )        /*  device.                   */
         OR (&M_GDEVIC EQ HPDRAFT) )      /*                            */
            %THEN %DO;                    /*                            */

                    GOPTIONS GSFMODE = APPEND;

                %END;                     /* End plotter specified      */
                                          /*  as graphics device.       */

        /*                                                               */
        /*    Output the final Ternary Plot on the specified            */
        /*        graphics device.                                      */
        /*                                                               */

        PROC GREPLAY IGOUT       = TERNPLOT
                     CC          = SASDATA.TERNCMAP
                     CMAP        = &M_COLMAP
                     NOFS;

        REPLAY TERNPLOT;

        RUN;

        QUIT;

        /*                                                               */
        /*    Reset all of the Titles, reset all of the Footnotes,      */
        /*        set the graphics options to the appropriate values    */
        /*        to prevent output on the specified graphics device,   */
        /*        and delete all of the temporary datasets utilized     */
        /*        in the Ternary Plotting System.                       */
        /*                                                               */

        TITLE1;

        FOOTNOTE1;

        GOPTIONS NODISPLAY
                 GSFMODE   = OFF;

        PROC DATASETS LIBRARY = WORK
                      MEMTYPE = ALL
                      NOLIST;

            DELETE TERNVERT
                   TERNDATA
                   TERNANNO
                   /
                     MT = DATA;

            DELETE TERNPLOT
                   /
                     MT = CATALOG;

        RUN;

        QUIT;

%MEND M_TPPLOT;

/*    M_TPSCAL.SAS                                                       */
/*                                                                       */
/*************************************************************************/
/*                                                                       */
/*    The SAS MACRO Program M_TPSCAL creates the final SAS               */
/*        Annotate Dataset by concatenating and scaling the             */
/*        data in all of the preliminary SAS Annotate                   */
/*        Datasets for the Ternary Plotting System.                     */
/*                                                                       */
/*    Programmed by Leon Fedenczuk and Mark Bercov on 91-01-24          */
/*        for Canadian Hunter Exploration Ltd.                          */
/*                                                                       */
/*************************************************************************/
/*                                                                       */

%MACRO M_TPSCAL;                          /*                            */

    /*                                                                   */
    /*    Create the final SAS Annotate Dataset by concatentating       */
    /*        and scaling the data in all of the preliminary SAS        */
    /*        Annotate Datasets.                                        */
    /*                                                                   */

    DATA TERNANNO
        (LABEL = 'Ternplot Final Scaled Annotation');

    ATTRIB TEXT      LABEL    = 'Annotation Text'
                     LENGTH   = $ 14
                     INFORMAT = $14.
                     FORMAT   = $14.;

    SET SASDATA.TERNTNGL

    %IF (&M_GRID   EQ YES)
        %THEN %DO;

                SASDATA.TERNGRID

            %END;

        TERNVERT
        TERNDATA;

    X       = &M_XBASE + (X        * &M_XSCAL);

    Y       = &M_YBASE + (Y        * &M_YSCAL);

    RUN;

%MEND M_TPSCAL;
-
```

```
/*    M_TPVERT.SAS                                                    */
/*                                                                    */
/********************************************************************/
/*                                                                    */
/*    The SAS MACRO Program M_TPVERT creates a SAS Annotate           */
/*        Dataset which contains the appropriate annotation           */
/*        output for the triangle vertices for the Ternary            */
/*        Plotting System.                                            */
/*                                                                    */
/*    Programmed by Leon Fedenczuk and Mark Bercov on 91-01-24        */
/*        for Canadian Hunter Exploration Ltd.                        */
/*                                                                    */
/********************************************************************/
/*                                                                    */

%MACRO M_TPVERT;                          /*                          */

    /*                                                                */
    /*    Create the SAS Annotate Dataset which contains the          */
    /*        appropriate annotation output for the triangle          */
    /*        vertices.                                               */
    /*                                                                */

    DATA TERNVERT
        (LABEL = 'Ternplot Triangle Vertex Annotation');

    %DCLANNO;                             /* Define length and data   */
                                          /*  type specifications     */
                                          /*  for annotate variables. */

    ATTRIB TEXT      LABEL    = 'Annotation Text'
                     LENGTH   = $ 14
                     INFORMAT = $14.
                     FORMAT   = $14.;

    ATTRIB TRIANHGT LABEL    =
                        'Relative Height of Triangle (%)
                     LENGTH   = 8
                     INFORMAT = 7.4
                     FORMAT   = 7.4;

    DROP TRIANHGT;

    TEXT     = '              ';

    TRIANHGT = (SQRT (0.75) * 100.0);   /* Calculate relative height  */
                                        /*  of triangle.              */

    %SYSTEM (3, 3, 3)                     /* Define all coordinates   */
                                          /*  as screen percentages.  */

    %IF (%BQUOTE (&M_XVLAB)  NE )         /* If label has been        */
        %THEN %DO;                        /*  specified for X-Vertex  */
                                          /*  of triangle, draw       */
                                          /*  X-Vertex label.         */

                %LABEL  (        0.0,        0.0,
                          "&M_XVLAB",      GREEN,
                                   0,          0,
                          &M_HVCHAR,       SWISS,
                                   8 )

            %END;                         /* End X-Vertex label.      */

    %IF (%BQUOTE (&M_YVLAB)  NE )         /* If label has been        */
        %THEN %DO;                        /*  specified for Y-Vertex  */
                                          /*  of triangle, draw       */
                                          /*  Y-Vertex label.         */

                %LABEL  (      100.0,        0.0,
                          "&M_YVLAB",      GREEN,
                                   0,          0,
                          &M_HVCHAR,       SWISS,
                                   8 )

            %END;                         /* End Y-Vertex label.      */

    %IF (%BQUOTE (&M_ZVLAB)  NE )         /* If label has been        */
        %THEN %DO;                        /*  specified for Z-Vertex  */

                                          /*  of triangle, draw       */
                                          /*  Z-Vertex label.         */

                %LABEL  (       50.0, TRIANHGT,
                          "&M_ZVLAB",      GREEN,
                                   0,          0,
                          &M_HVCHAR,       SWISS,
                                   2 )

            %END;                         /* End Z-Vertex label.      */

    RUN;

%MEND M_TPVERT;
```