DEPARTMENT OF THE INTERIOR

U.S. GEOLOGICAL SURVEY

A Simple Ternary/Quaternary Plotting Program for Microcomputers

by

J. T. O'Connor[1]

Open-File Report 87-171

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards.

[1]U.S. Geological Survey
Reston, VA

A SIMPLE TERNARY/QUATERNARY PLOTTING PROGRAM FOR MICROCOMPUTERS

by J.T. O'Connor
(Branch of Sedimentary Processes
U.S.Geological Survey)

INTRODUCTION

The earth-scientist frequently finds a need for displaying intermediate and final petrographic and geochemical data. This report describes a simple computer program, **Petplot**, developed for microcomputers using MS-DOS that creates CRT displays of values for one to four components on a ternary or quaternary diagram. The program also produces two data files as output that can be used to make camera-ready illustrations of the results. The program requires a microcomputer with graphics capability; it was developed on a Compaq 286 with a Quadram EGA+ graphics card. The program is written in TurboPascal using the 320 x 200 colorgraphmode. The object code requires an 8087 (or equivalent) chip for operation. The source code can be recompiled on a machine without this chip and color capability is not necessary for its use.

PROGRAM DESIGN

Petplot is similar to and draws on plotting programs developed for other research purposes (Schachte et al, 1986; Beaudoin and Bowyer-Beaudoin, 1981; and Armienti, 1986) but is oriented toward the easy use of an MS-DOS microcomputer as an aid to data analysis during data collection. The program uses standard methods of rotating points and shapes in space (Artwick, 1985) and standard matrix operations developed for the Pascal language (Crandall, 1984).

Because the specific components that will be important in the user's analysis are not always known in the early stages of a study, the CRT screen display was made value-free and a user-choice of ternary or quaternary projection was incorporated into the design. While this choice was designed to be appropriate for petrographic or geochemical applications, any closed system of one to four components may be displayed. Speed and simplicity of operation are two criteria that determine the value of a display program in helping to assess an analysis in progress; therefore, the screen display and a simple input file structure was utilized. Plotting options include value-free plots, generalized plots that can be edited to specific needs, and a choice of sedimentary petrologic classification overlays. The Dynamic Graphics Interactive Surface Modeling (ISM) file-structure was chosen as the final output file format directed toward plotting camera-ready illustrations; modification of the source code to produce output files compatible with other plotting software should be a trivial task for the interested user.

Data read into the program for four components are first normalized creating a closed system. If a ternary projection of the data is desired, components three and four are added together; therefore, a zero value for component four should be entered if only three values are to be considered.

For the ternary diagram, the cartesian horizontal (h) and vertical (v) coordinates of the individual points are calculated from the normalized values in terms of screen-parameters:

$$h = 20 + (250(1-B)\cos(30) - 250\cos(30)A/2)/2$$

$$v = 180 - 165A$$

where A and B are the components to be represented by the apex and the lower left hand corners of the diagram, respectively.

For the quaternary diagram, the orthogonal x, y, and z values of a point are calculated by solution of the matrix equation:

$$\begin{vmatrix} 0 & .33 & .94 \\ 0 & 1 & 0 \\ .82 & -.33 & .47 \end{vmatrix} = \begin{vmatrix} .82(1-A) \\ .82B \\ .82C \end{vmatrix}$$

where B is the component at the apex of the tetrahedron and A and C are the components at the lower front and lower right hand corners, respectively.

The Cartesian screen-oriented coordinates of the points are calculated from the x and y values obtained above by:

$h = 160 + 200x$

$v = 170 - 200y$

The apices of the tetrahedron are calculated by assuming A, B, C, and D successively equal to 1. The original projection assumes zero degrees rotation and portrays a tetrahedron with a horizontal base and one apex centered in this plane pointing toward the observer. User-selected rotations of the tetrahedron about the X-axis are positive if the forward apex rises on the screen, negative if it falls. Rotations about the Y-axis are positive if the forward apex moves right on the screen, negative if it moves left.

The tetrahedron, and the points within it, are rotated by multiplying the (x,y,z) vector first by an X-axis rotation matrix, then by a Y-axis rotation matrix:

$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos(rx) & -\sin(rx) \\ 0 & \sin(rx) & \cos(rx) \end{vmatrix} \; X \; \begin{vmatrix} x \\ y \\ z \end{vmatrix} = \begin{vmatrix} x_1 \\ y_1 \\ z_1 \end{vmatrix}$$

&

$$\begin{vmatrix} \cos(ry) & 0 & -\sin(ry) \\ 0 & 1 & 0 \\ \sin(ry) & 0 & -\cos(ry) \end{vmatrix} \; X \; \begin{vmatrix} x_1 \\ y_1 \\ z_1 \end{vmatrix} = \begin{vmatrix} x_2 \\ y_2 \\ z_2 \end{vmatrix}$$

where (rx) and (ry) are the user-selected X-axis and Y-axis rotation angles, respectively.

Lines are drawn between the apices of the tetrahedron and points plotted (on the screen) as small circles of (two) different colors. Labels are not included on the screen display because they would cause unnecessary delay in program operation and the screen print is non-transferable. Appropriate labels can be added to the finished illustration by modifying the output file using an editor.

PROGRAM OPERATION

This part of the report is a description of some of the interactive portions of the program operation. A step-by-step Userguide is comprises the following part.

Input data for the program should reside in a file with a conventional DOS path and filename of 16 or fewer characters (combined). The file should consist of one line per point. Each line will consist of an ID, four values separated by a comma or spaces and an integer value representing the color to be displayed for a group of points **(Figure 1)**. While a registered columnar format is helpful in correcting data, it is not necessary for the program to function. In the present version of **Petplot** the colornumber 0 will not be displayed, 3 will plot yellow, 2 will plot red and 1 will plot green. The color numbers are converted to other symbols for the ISM output files (3 = x, 2 = *, 1 = o).

Response to the program's queries has been kept simple for speed and ease of use. No error-recognition has been built into the interactive part of the program, however, so entry of typos or other bad values for file names will generally result in a program crash and a need to restart the program from the beginning. Typos in the data files will plot incorrectly.

The screen display is developed for quick perusal of data during collection. Because of disparities between horizontal and vertical units on the CRT display, distortions of the figures may result. The files generated for plotting do not reflect these distortions, thus should represent true, scalable values for components.

Two ISM files are developed during a run of **Petplot**; a scattered data file (scatdat.dat) and an annotation file (plotfile.dat). The scattered data file (**Figure 2**) is created to provide a blank "map" on which the ISM annotation file is superposed to create the illustration. The scattered data file provides the x and y coordinates of the figure, modified to allow fitting of the legend, and constant z coordinates that will suppress ISM's attempts to contour the resulting "map".

The annotation file (**Figure 3**) is a symbolic representation of the investigator's data; it will cause one of several triangular diagrams (**Figures 4, 5, 6 and 7**) or the tetrahedron (**Figure 8**) to be drawn and the points to be plotted - using different symbols for the different colors.

Either of these files may be readily modified using an editor to change symbols, to delete samples, or to change or add labels. To change the labels on the illustration the annotation file is modified by replacing a specific label (**e.g.** "QUATERNARY DIAGRAM","TITLE AREA" with the desired label (**e.g.** "KANAWHA/POTTSVILLE","SANDSTONE COMPOSITIONS", **etc.**) and running the ISM program to produce a plotfile and subsequent illustration (**Figure 9**). Symbols can be modified with the aid of ISM documentation by changing the appropriate SRFSYM value for the quaternary plot or the character in the quotes of the SRFTXT command for a ternary plot. New samples must be added to the input file to preserve the rotational continuity of the projections.

Because the files produced by **Petplot** are named within the program, the user must rename them or remove them to a different directory before reusing **Petplot**. Rerunning the program will immediately write over existing data.

The program is designed to show the back edges of the tetrahedron as dashed lines in the final illustration (ISM plot). For simplicity, this phase of the program uses an approximation of ± 30 degrees for the y-axis rotation that would cause the appropriate edges to be hidden. The actual visibility of these edges is also dependent upon the x-axis rotation; therefore, the edges may show up as hidden when they are not if the y-axis rotation is close to 32 degrees (± 2 degrees) and the x-axis rotation is ($\pm$)0-10 degrees.

USERGUIDE FOR PETPLOT

Keyboard Entry	Screen Response/(*) Remarks
1.	A> (*) System prompt
2. petplot	Enter the name of the input file (eg. b:\potts.dat):
3. <filename>	Do you wish: 1. Ternary plot of elements 1, 2, & 3 + 4. 2. Quaternary plot of elements 1, 2, 3, 4. 3. Exit Please enter 1, 2, or 3.
4a. 1	(*) Ternary plot of the data from the user's file with the notation: Press any key to continue
5a. <any key>	Do you wish your illustration to include: 1. Generalized Titles & Tic Marks. 2. McBride, 1963, Sedimentary Class. Overlay 3. McBride, 1963, Fields Labeled, No Points. 4. Folk, 1980, Sedimentary Class. Overlay 5. No Titles, No Tic Marks Please enter a number (1-5):
6a. <1-5>	ISM annotation file of data written as "plotfile.dat". ISM scattered data file of outline boundaries written as "scatdat.dat".

Keyboard Entry		**Screen Response**/(*) Remarks
4b.	2	(*) Quaternary plot of the data from the user's file with zero rotation about X-axis or Y-axis and the notation: **Do you wish** **another** **viewing angle?** **(Enter y or n)**
5b.	y	**Current x-rotation = 0 deg.** **Enter new x-rotation:**
6b.	<any number>	**Current y-rotation = 0 deg.** **Enter new y-rotation:**
7b.	<any number>	(*) Quaternary plot of the data from the user's file with the new rotation angles and the notation: **Do you wish** **another** **viewing angle?** **(Enter y or n)** (*) Another angle may be selected or the user may exit by entering n.
8b.	n	**ISM annotation file of data written as "plotfile.dat".** **ISM scattered data file of outline boundaries written as "scatdat.dat".** **C>** (*) System prompt
4c.	3	**C>** (*) System prompt

SOURCE CODE

The following source code was written in TurboPascal for operation using MS-DOS.

```
program pettest;
{$I GRAPH.P}

label
   newview;

type
   matrix4x3 = array[1..4,1..3] of real;
   matrix3x3 = array[1..3,1..3] of real;

var
   labnum:                                       string[8];
   x,y,z1,z2,rotatx,rotaty:                      real;
   infilename:                                   string[16];
   infile, outfile, outfile2:                    text;
   ans1:                                         char;
   ans0:                                         integer;

procedure gloss;

var
   x1 ,x2, x3, y1, y2, y3:                       real;
   ans2, i, j:                                   integer;

begin
   textmode(3); textbackground(cyan); window(9,6,62,15);
clrscr;
   writeln; writeln(' Do you wish your illustration to
include:');
   writeln('    1. Generalized Titles & Tic Marks.');
   writeln('    2. McBride, 1963, Sedimentary Class.
Overlay.');
   writeln('    3. McBride, 1963, Fields Labeled, No
Points.');
   writeln('    4. Folk, 1980, Sedimentary Class. Over-
lay.');
   writeln('    5. No Titles, No Tic Marks.');
   writeln; textcolor(red);
   writeln(' Please enter a number (1-5):');
   readln(ans2);
   if ans2 = 3 then rewrite(outfile);

   {write ternary outline to ISM annotation file.}
```

```
   writeln(outfile, 'SETPEN 1,1,1');
   writeln(outfile, 'SETTXT 3,4,1.0,2');
   writeln(outfile, 'LGDFRM 0,0,2');
   writeln(outfile, 'LGDBOX -9,3,30.0%,25.0%,0,0');
   writeln(outfile, 'SRFLNE 2,2,0');
   writeln(outfile, '       ', '120.0, 173.2');
   writeln(outfile, '       ', ' 20.0,   0.0');
   writeln(outfile, '       ', '220.0,   0.0');
   writeln(outfile, '       ', '120.0, 173.2');

   {select general titles & tic marks}
   if ans2 = 1 then
   begin
      writeln(outfile, 'LGDTXT 2," ","TERNARY
DIAGRAM","TITLE AREA"');
      writeln(outfile, 'SETTXT 3,3,1.0,2');
      writeln(outfile, 'LGDTXT 2," "," "');
      writeln(outfile, 'LGDSYM 2,2,3,4,"Group 1"');
      writeln(outfile, 'LGDSYM 2,13,3,4,"Group 2"');
      writeln(outfile, 'LDGSYM 2,3,3,4,"Group 3"');
      writeln(outfile, 'SRFTXT 2,0');
      writeln(outfile, '       ', '120.0  176.0  "PHASE
A"');
      writeln(outfile, 'SRFTXT 4,0');
      writeln(outfile, '       ', '223.0    0.0  "PHASE
B"');
      writeln(outfile, 'SRFTXT 6,0');
      writeln(outfile, '       ', ' 17.0    0.0  "PHASE
C"');
      for i := 1 to 9 do
         for j := 1 to 3 do
         begin
            if j = 1 then
            begin
               x1 := 19 + i*20; y1 := -1.73;
               x2 := 20 + i*20; y2 := 0.0;
               x3 := 21 + i*20; y3 := -1.73;
            end;
            if j = 2 then
            begin
               x1 := 18 + i*10; y1 := i*17.32;
               x2 := 20 + i*10; y2 := i*17.32;
               x3 := 19 + i*10; y3 := i*17.32 + 1.73;
            end;
            if j = 3 then
            begin
               x1 := 222 - i*10; y1 := i*17.32;
               x2 := 220 - i*10; y2 := i*17.32;
```

```
                x3 := 221 - i*10; y3 := i*17.32 + 1.73;
            end;
            writeln(outfile, 'SRFLNE 1,1,0');
            writeln(outfile, '        ',x1:5:1, ',
',y1:5:1);
            writeln(outfile, '        ',x2:5:1, ',
',y2:5:1);
            writeln(outfile, '        ',x3:5:1, ',
',y3:5:1);
         end;
      writeln(outfile, 'SETTXT 1,2,0.5,3');
   end;

   {select McBride classification overlay of data
points}
   if (ans2 = 2) or (ans2 = 3) then
   begin
      writeln(outfile,
      'LGDTXT 2," ","CLASSIFICATION OF","SANDSTONES"');
      writeln(outfile, 'SETTXT 3,3,1.0,2');
      writeln(outfile, 'LGDTXT 2,"from McBride, 1963"');
      writeln(outfile, 'LGDTXT 2," "," "');
      writeln(outfile, 'LGDSYM 2,2,3,4,"Group 1"');
      writeln(outfile, 'LGDSYM 2,13,3,4,"Group 2"');
      writeln(outfile, 'SRFTXT 2,0');
      writeln(outfile, '        ', '120.0  176.0  "QUARTZ
+ CHERT"');
      writeln(outfile, 'SRFTXT 4,0');
      writeln(outfile, '        ', '223.0    5.0  "UN-
STABLE"');
      writeln(outfile, '        ', '223.0    0.0
"LITHIC"');
      writeln(outfile, '        ', '223.0   -5.0  "FRAG-
MENTS"');
      writeln(outfile, 'SRFTXT 6,0');
      writeln(outfile, '        ', ' 17.0    0.0  "FELD-
SPAR"');
      writeln(outfile, 'SRFLNE 4,1,0');
      writeln(outfile, '        ', '200.0,   0.0
"first"');
      writeln(outfile, '        ', '120.0, 138.6');
      writeln(outfile, '        ', ' 40.0,   0.0');
      writeln(outfile, '        ', ' 95.0, 129.9   "sec-
ond"');
      writeln(outfile, '        ', '120.0,  86.6');
      writeln(outfile, '        ', '145.0, 129.9');
      writeln(outfile, '        ', '115.0, 164.5
"third"');
```

```
      writeln(outfile, '         ', '120.0, 155.9');
      writeln(outfile, '         ', '125.0, 164.5');
      writeln(outfile, '         ', '120.0, 155.9
"fourth"');
      writeln(outfile, '         ', '120.0, 138.6');
      writeln(outfile, '         ', '120.0,  86.6
"fifth"');
      writeln(outfile, '         ', '120.0,   0.0');
      writeln(outfile, 'SRFTXT 4,0');
      writeln(outfile, '         ', '125.0, 164.5    "
5%"');
      writeln(outfile, '         ', '145.0, 129.9    "
25%"');
      writeln(outfile, '         ', '170.0,  86.6    "
50%"');
      writeln(outfile, '         ', '195.0,  43.3    "
25%"');
      writeln(outfile, 'SRFTXT 6,0');
      writeln(outfile, '         ', '114.0, 164.5    "5%
"');
      writeln(outfile, '         ', ' 94.0, 129.9    "25%
"');
      writeln(outfile, '         ', ' 69.0,  86.6    "50%
"');
      writeln(outfile, '         ', ' 44.0,  43.3    "25%
"');
      writeln(outfile, 'SRFTXT 8,0');
      writeln(outfile, '         ', ' 40.0,  -1.0
"10%"');
      writeln(outfile, '         ', ' 60.0,  -1.0
"25%"');
      writeln(outfile, '         ', '120.0,  -1.0
"50%"');
      writeln(outfile, '         ', '180.0,  -1.0
"25%"');
      writeln(outfile, '         ', '200.0,  -1.0
"10%"');
   end;

   {select McBride classification diagram, no data
points}
   if ans2 = 3 then
   begin
      writeln(outfile, 'SRFTXT 8,0');
      writeln(outfile, '         ', ' 91.0,  43.3
"Lithic"');
      writeln(outfile, '         ', ' 91.0,  38.3
"arkose"');
```

```
      writeln(outfile, '          ', '145.0,   43.3    "Feld-
spathic"');
      writeln(outfile, '          ', '145.0,   38.3    "lith-
arenite"');
      writeln(outfile, '          ', '120.0, 116.0
"Lithic"');
      writeln(outfile, '          ', '120.0, 111.0    "sub-
arkose"');
      writeln(outfile, 'SRFTXT 4,0');
      writeln(outfile, '          ', '125.0, 169.0    "Quart-
zarenite"');
      writeln(outfile, 'SRFTXT 4,60');
      writeln(outfile, '          ', ' 62.0,   59.0
"Arkose"');
      writeln(outfile, '          ', '103.0, 127.0    "Sub-
arkose"');
      writeln(outfile, 'SRFTXT 4,-60');
      writeln(outfile, '          ', '126.5, 149.5
"Sublith-"');
      writeln(outfile, '          ', '123.5, 147.5
"arenite"');
      writeln(outfile, '          ', '162.0,   81.0    "Lith-
arenite"');
   end;

   {select Folk classification overlay on data points}
   if ans2 = 4 then
   begin
      writeln(outfile,
      'LGDTXT 2," ","CLASSIFICATION OF","SANDSTONES"');
      writeln(outfile, 'SETTXT 3,3,1.0,2');
      writeln(outfile, 'LGDTXT 2,"from Folk, 1980"');
      writeln(outfile, 'LGDTXT 2," "," "');
      writeln(outfile, 'LGDSYM 2,2,3,4,"Group 1"');
      writeln(outfile, 'LGDSYM 2,13,3,4,"Group 2"');
      writeln(outfile, 'SRFTXT 2,0');
      writeln(outfile, '          ', '120.0  176.0
"QUARTZ"');
      writeln(outfile, 'SRFTXT 4,0');
      writeln(outfile, '          ', '223.0    2.5
"ROCK"');
      writeln(outfile, '          ', '223.0   -2.5   "FRAG-
MENTS"');
      writeln(outfile, 'SRFTXT 6,0');
      writeln(outfile, '          ', ' 17.0    0.0   "FELD-
SPAR"');
      writeln(outfile, 'SRFLNE 4,1,0');
      writeln(outfile, '          ', '115.0, 164.5
"first"');
```

```
      writeln(outfile, '          ', '125.0, 164.5');
      writeln(outfile, '          ', '120.0, 164.5   "sec-
ond"');
      writeln(outfile, '          ', '120.0,   0.0');
      writeln(outfile, '          ', ' 95.0, 129.0
"third"');
      writeln(outfile, '          ', '145.0, 129.0');
      writeln(outfile, '          ', ' 70.0,   0.0
"fourth"');
      writeln(outfile, '          ', '107.5, 129.0');
      writeln(outfile, '          ', '170.0,   0.0
"fifth"');
      writeln(outfile, '          ', '132.5, 129.0');
      writeln(outfile, 'SETTXT 1,2,0.5,3');
      writeln(outfile, 'SRFTXT 8,0');
      writeln(outfile, '          ', ' 70.0,  -1.0
"3:1"');
      writeln(outfile, '          ', '120.0,  -1.0
"1:1"');
      writeln(outfile, '          ', '170.0,  -1.0
"1:3"');
      writeln(outfile, '          ', '120.0,  -6.0   "F/R
Ratio"');
      writeln(outfile, 'SRFTXT 6,0');
      writeln(outfile, '          ', ' 94.0, 129.0   "75%
"');
      writeln(outfile, '          ', '114.0, 164.5   "95%
"');
   end;
end;

procedure ternplot(var x,y,z1,z2: real);

var
   normtot, normal,s, hh, vv:                  real;
   h,v,colornum, i, j:                         integer;
   labnum:                                     string[8];

begin
   normal := cos(30*3.1416/180); s := 200;

   {write general boundaries for ternary plot to
   ISM scattered data file}
   writeln(outfile2,' 10.0   180.0   1');
   writeln(outfile2,'230.0   180.0   1');
   writeln(outfile2,'145.0   -10.0   1');

   {read data point-by-point.}
```

```
    while not EOF(infile) do
    begin
       readln (infile,labnum,x,y,z1,z2,colornum);
       normtot := x+y+z1+z2;
       x := x/normtot;
       y := y/normtot;
       v := 180 - round(x*132); vv := 173.205*x;
       h := 20 + round(((1-y)*s*normal -
x*s*normal/2)/normal);
       circle(h,v,1,colornum);

       {write points to ISM annotation file}
       if colornum > 0 then
       begin
          hh := h/1.0;
          writeln(outfile, 'SRFTXT 5,0');
          if colornum = 1 then
             writeln(outfile, '        ',hh:5:1,',
',vv:5:1,
                         '      "o"');
          if colornum = 3 then
             writeln(outfile, '        ',hh:5:1,',
',vv:5:1,
                         '      "x"');
          if colornum = 2 then
             writeln(outfile, '        ',hh:5:1,',
',vv:5:1,
                         '      "*"');
       end;
    end;
    writeln('Press any key');
    writeln('to continue.');
    Draw(20,180,220,180,2);
    Draw(220,180,120,48,2);
    Draw(120,48,20,180,2);
    repeat until keypressed;
    gloss;
end;

procedure multmatrix(n,m,k: integer;
          rmx,rmy: matrix3x3;
          var rmt, rm3: matrix4x3);

var
   i,j,h,ii,jj,hh:                                    integer;

begin
   {y-axis rotation; result in rm3[i,j]}
```

```
    for i := 1 to n do
    begin
       for j := 1 to k do
       begin
          rm3[i,j] := 0;
          for h := 1 to m do
             rm3[i,j] := rm3[i,j] + rmt[i,h]*rmy[h,j];
       end;
    end;
    {x-axis rotation; result in rmt[ii,jj]}
    for ii := 1 to n do
    begin
       for jj := 1 to k do
       begin
          rmt[ii,jj] := 0;
          for hh := 1 to m do
             rmt[ii,jj] := rmt[ii,jj] +
rm3[ii,hh]*rmx[hh,jj];
       end;
    end;
end;

procedure pointsolv(var rmpt: matrix3x3; var rmt:
matrix4x3);

var
   k : integer;
   d: real;

   function determ(rmpt: matrix3x3): real;

   var
      fac1,fac2,fac3:                         real;

   begin
      fac1 := rmpt[1,1]*(rmpt[2,2]*rmpt[3,3]-
rmpt[3,2]*rmpt[2,3]);
      fac2 := rmpt[1,2]*(rmpt[2,1]*rmpt[3,3]-
rmpt[3,1]*rmpt[2,3]);
      fac3 := rmpt[1,3]*(rmpt[2,1]*rmpt[3,2]-
rmpt[3,1]*rmpt[2,2]);
      determ := fac1-fac2+fac3;
   end;

   procedure swap(k: integer; var rmpt: matrix3x3;
              var rmt: matrix4x3);

   var
```

```
      e:                                                  real;
      j:                                                  integer;

   begin
      for j := 1 to 3 do
      begin
         e := rmt[1,j];
         rmt[1,j] := rmpt[j,k];
         rmpt[j,k] := e;
      end;
   end;

begin
   d := determ(rmpt);
   for k := 1 to 3 do
   begin
      swap(k,rmpt,rmt);
      rmt[2,k] := determ(rmpt)/d;
      swap(k,rmpt,rmt);
   end;
end;

procedure ism_tetrahedron(s, rotatx, rotaty: real; rmt:
matrix4x3);

var
   i:                                                    integer;
   miny:                                                 real;

begin
   {write tetrahedron outline to ISM annotation file}
   writeln(outfile, 'SETPEN 1,1,1');
   writeln(outfile, 'SETTXT 3,4,1.0,2');
   writeln(outfile, 'LGDFRM -1,-1,2');
   writeln(outfile, 'LGDBOX -9,3,30.0%,25.0%,0,0');
   writeln(outfile, 'LGDTXT 2," ","QUATERNARY
DIAGRAM","TITLE AREA"');
   writeln(outfile, 'SETTXT 3,3,1.0,2');
   writeln(outfile, 'LGDTXT 2," "," "');
   writeln(outfile, 'LGDSYM 2,2,3,4,"Group 1"');
   writeln(outfile, 'LGDSYM 2,13,3,4,"Group 2"');
   writeln(outfile, 'LGDSYM 2,3,3,4,"Group 3"');
   writeln(outfile, 'SRFLNE 2,2,0');
   writeln(outfile, '        ',
s*rmt[1,1]:8:2,',',s*rmt[1,2]:8:2);
   writeln(outfile, '        ',
s*rmt[2,1]:8:2,',',s*rmt[2,2]:8:2);
   if rotaty <-30.0*3.1416/180 then
```

```
   begin
      writeln(outfile, 'SRFLNE 3,2,0');
      writeln(outfile, '        ',
s*rmt[2,1]:8:2,',',s*rmt[2,2]:8:2);
   end;
   writeln(outfile, '        ',
s*rmt[3,1]:8:2,',',s*rmt[3,2]:8:2);
   writeln(outfile, '        ',
s*rmt[1,1]:8:2,',',s*rmt[1,2]:8:2);
   if rotaty >30.0*3.1416/180 then writeln(outfile,
'SRFLNE 3,2,0')
      else writeln(outfile, 'SRFLNE 2,2,0');
   writeln(outfile, '        ',
s*rmt[1,1]:8:2,',',s*rmt[1,2]:8:2);
   writeln(outfile, '        ',
s*rmt[4,1]:8:2,',',s*rmt[4,2]:8:2);
   writeln(outfile, '        ',
s*rmt[2,1]:8:2,',',s*rmt[2,2]:8:2);
   if rotatx < 0 then writeln(outfile, 'SRFLNE 3,2,0')
      else writeln(outfile, 'SRFLNE 2,2,0');
   writeln(outfile, '        ',
s*rmt[4,1]:8:2,',',s*rmt[4,2]:8:2);
   writeln(outfile, '        ',
s*rmt[3,1]:8:2,',',s*rmt[3,2]:8:2);
   writeln(outfile, 'SRFTXT 2,0');
   writeln(outfile, '        ',
s*rmt[2,1]:8:2,',',3+s*rmt[2,2]:8:2,
           ' "PHASE A"');
   if rotaty <-30*3.1416/180 then writeln(outfile,
'SRFTXT 4,0')
      else writeln(outfile, 'SRFTXT 8,0');
   writeln(outfile, '        ', 3+s*rmt[3,1]:8:2,',',-
3+s*rmt[3,2]:8:2,
           ' "PHASE C"');
   if rotaty >30*3.1416/180 then writeln(outfile,
'SRFTXT 6,0')
      else writeln(outfile, 'SRFTXT 8,0');
   writeln(outfile, '        ',s*rmt[4,1]-3:8:2,',',-
3+s*rmt[4,2]:8:2,
           ' "PHASE D"');
   writeln(outfile, 'SRFTXT 8,0');
   writeln(outfile, '
',s*rmt[1,1]:8:2,',',s*rmt[1,2]-3:8:2,
           ' "PHASE B"');
   writeln(outfile,'SETTXT 1,2,0.5,3');

   {write tetrahedron boundaries to ISM scattered data
file}
```

```
   for i := 1 to 4 do if rmt[i,2] < miny then miny :=
rmt[i,2];
   writeln(outfile2, ' -115.0,  ',-10.0+s*miny:4:1, ',
1');
   writeln(outfile2, ' -115.0,  162.0,  1');
   writeln(outfile2, '  125.0,  ',-10.0+s*miny:4:1, ',
1");
   writeln(outfile2, '  125.0,  162.0,  1');
   writeln(outfile2, '    0.0,   75.0   1');
   end;

procedure quatplot(var x,y,z1,z2,rotatx,rotaty: real);

var
   labnum:                                      string[8];
   s, sum:                                      real;
   i,j,h,n,m,k,
   h0,v0,h1,v1,h2,v2,colornum:                  integer;
   rmt:                                         matrix4x3;
   rmx: {rotation matrix for x axis}            matrix3x3;
   rmy: {rotation matrix for y axis}            matrix3x3;
   rm3: {matrix representing rotated point}     matrix4x3;
   rmpt:{coefficient matrix}                    matrix3x3;

begin
   S := 200;
   {set coordinates of unrotated tetrahedron}
   {   x                 y                 z
}
{A} rmt[1,1] := 0; rmt[1,2] := 0; rmt[1,3] := 0 -
0.57735;
{B} rmt[2,1] := 0; rmt[2,2] := 0.86603*0.94281; rmt[2,3]
:= 0;
{C} rmt[3,1] := 0.5; rmt[3,2] := 0; rmt[3,3] :=
0.33333*0.86603;
{D} rmt[4,1] := -0.5; rmt[4,2] := 0; rmt[4,3] :=
0.33333*0.86603;
   {set values for rotation matrices}
   {x-axis rotation}
   rmx[1,1] := 1; rmx[1,2] := 0; rmx[1,3] := 0;
   rmx[2,1] := 0; rmx[2,2] := cos(rotatx); rmx[2,3] := -
sin(rotatx);
   rmx[3,1] := 0; rmx[3,2] := sin(rotatx); rmx[3,3] :=
cos(rotatx);
   {y-axis rotation}
   rmy[1,1] := cos(rotaty); rmy[1,2] := 0; rmy[1,3] :=
sin(rotaty);
   rmy[2,1] := 0; rmy[2,2] := 1; rmy[2,3] := 0;
```

```
   rmy[3,1] := sin(rotaty); rmy[3,2] := 0; rmy[3,3] := -
cos(rotaty);

   {rotate tetrahedron}
   multmatrix(4,3,3,rmx,rmy,rmt,rm3);

   {write tetrahedron coordinates to ISM output files.}
   ism_tetrahedron(s, rotatx, rotaty, rmt);

   {draw tetrahedron on screen.}
   h1 := round(160 + s*rmt[1,1]); v1 := round(170 -
s*rmt[1,2]);
   for i := 2 to 4 do
   begin
      h2 := round(160 + s*rmt[i,1]);
      v2 := round(170 - s*rmt[i,2]);
      draw(h1,v1,h2,v2,2);
   end;
   h1 := round(160 + s*rmt[2,1]); v1 := round(170 -
s*rmt[2,2]);
   for i := 3 to 4 do
   begin
      h2 := round(160 + s*rmt[i,1]);
      v2 := round(170 - s*rmt[i,2]);
      draw(h1,v1,h2,v2,2);
   end;
   h1 := round(160 + s*rmt[3,1]); v1 := round(170 -
s*rmt[3,2]);
   draw(h1,v1,h2,v2,2);
   while not EOF(infile) do  {Read data point-by-point}
   begin
      readln(infile,labnum,y,x,z1,z2,colornum);
      sum := x + y + z1 + z2;

      {Set translation determinant values}
      {rmt[2,i] is used as the unknown point vector}
      {rmt[1,i] is used as the constant vector}

      rmpt[2,1] := 0; rmpt[2,2] := 1; rmpt[2,3] := 0;
      rmpt[1,1] := 0; rmpt[1,2] := 0.33333; rmpt[1,3] :=
0.94281;
      rmpt[3,1] := 0.8165; rmpt[3,2] := -0.33333;
      rmpt[3,3] := 0.4714;
      rmt[1,1] := 0.86603*0.94281*(1-z1/sum);
                      {amount of component A}
      rmt[1,2] := 0.86603*0.94281*y/sum; {amount of com-
ponent B}
      rmt[1,3] := 0.86603*0.94281*(z2/sum);
```

```
                    {amount of component C}

        {solve for the cartesian coordinates of the
original point}
        pointsolv(rmpt,rmt);

        {rotate the point}
        for i := 1 to 3 do rmt[1,i] := rmt[2,i]; rmt[1,3]
:= rmt[1,3]-0.57735;
        multmatrix(1,3,3,rmx,rmy,rmt,rm3);
        {writeln(lst,'Rotated point coordinates: x =
',rmt[1,1],', y = ',
              rmt[1,2],', z =',rmt[1,3]);}

        {draw the point}
        h1 := round(160 + s*rmt[1,1]);
        v1 := round(170 - s*rmt[1,2]);
        circle(h1,v1,1,colornum);

        {write point coordinates to ISM output (annota-
tion) file.}
        if colornum > 0 then
        begin
           if colornum = 1 then colornum := 13;
           writeln(outfile, 'SRFSYM ',colornum,', 2,1,0');
           writeln(outfile, '        ', s*rmt[1,1]:8:2,',',
s*rmt[1,2]:8:2);
        end;
    end;
end;

begin
    clrscr;
    textmode(3); textbackground(9); window(9,9,62,12);
clrscr;
    writeln;
    writeln(' Enter the name of the input file (eg.
b:\potts.dat):');
    readln(infilename); clrscr;
    assign(infile, infilename); assign(outfile, 'plot-
file.dat');
    assign(outfile2, 'scatdat.dat');
    rotatx := 0; rotaty := 0;
    textmode(3); textbackground(9); window(10,9,62,16);
clrscr;
    writeln; writeln('  Do you wish:');
    writeln('      1.  Ternary plot of elements 1, 2, & 3
+ 4.');
```

```
   writeln('      2.  Quaternary plot of elements 1, 2,
3, 4.');
   writeln('      3.  Exit.'); textcolor(12);
   writeln; writeln('  Please enter 1, 2, or 3.');
   readln(ans0);
   if ans0 = 3 then exit;
   textmode(3);
   newview: reset(infile); rewrite(outfile);
rewrite(outfile2);
   GraphColorMode;
   {HiRes;}
   {HiResColor(12);}
   Palette(2);
   GraphBackground(0);
   ColorTable(0,1,2,3);
   if ans0 = 1 then ternplot(x,y,z1,z2);
   if ans0 = 2 then
   begin
      writeln;
      writeln(' Do you wish');
      writeln(' another');
      writeln(' viewing angle?'); writeln; writeln('
(Enter y or n)');
      draw(0,0,120,0,2);
      draw(120,0,120,60,2);
      draw(120,60,0,60,2);
      draw(0,60,0,0,2);
      quatplot(x,y,z1,z2,rotatx,rotaty);
      readln (ans1);
      if (ans1 = 'y') or (ans1 = 'Y') then
      begin
         textmode(3); textbackground(9);
window(2,1,32,7); clrscr;
         writeln; writeln;
         writeln(' Current x-rotation =
',(180/3.1416)*rotatx:3:0,' deg.');
         writeln(' Enter new x-rotation: ');
         readln(rotatx); clrscr;
         rotatx := rotatx*3.1416/180;
         writeln; writeln;
         writeln(' Current y-rotation =
',(180/3.1416)*rotaty:3:0,' deg.');
         writeln(' Enter new y-rotation: ');
         readln(rotaty);
         rotaty := rotaty*3.1416/180;
         goto newview;
      end;
   end;
```

```
     textmode(3); textbackground(9); window(20,9,50,16);
clrscr;
     writeln; writeln('  ISM annotation file of data');
     writeln('  written as "plotfile.dat".');
     writeln; writeln('  ISM scattered data file');
     writeln('  of outline boundaries');
     writeln('  written as "scatdat.dat".');
     close(infile); close(outfile); close(outfile2);
end.
```

REFERENCES

Armienti, P. 1986, **TETRASEZ: an interactive program in BASIC to perform tetrahedral diagrams.** *Computers & Geosciences,* vol. 12, pp. 229-241.

Artwick, B.A., 1985, **Chapter 7: Mathematics and Transforms for Advanced Graphics,** *Microcomputer Displays, Graphics, and Animation*, Prentice-Hall, Englewood Cliffs, N.J., pp. 194-211.

Beaudoin, Y., & Bowyer-Beaudoin, A., 1981, **SOILTD: A FORTRAN subroutine to plot textural data on a triangular diagram using an X-Y plotter**, *Computers & Geosciences*, vol. 7, pp. 207-214.

Crandall, R.E., 1984, *PASCAL Applications for the Sciences - A Self-Teaching Guide*, John Wiley & Sons, Inc., New York, 246 p.

Folk, R.L., 1980, *Petrology of Sedimentary Rocks*, Hemphill Publishing Co., Austin, Texas, 182 p.

McBride, E.F., 1963, **A Classification of Common Sandstones.** *Jour. Sed. Petrol.,* vol. 33, pp664-669.

Schachte, B.R., Pierce, B.S., & Johnson, M.F., 1986, **TRIANGL: A Ternary Diagram Program on the PRIME** Com**puter**, *U.S. Geol. Survey Open File Report 86-627*, 16 p.

ft-8b	96.8	0.4	2.2	0.0	0
ft-16k	83.8	3.2	10.8	0.6	3
ft-16c	58.2	11.6	20.2	2.4	3
ft-16b	43.5	9.2	27.6	0.2	3
ft-21c	93.8	0.0	1.8	0.0	1
ft-22	98.6	0.2	0.8	0.0	1
ft-16e	64.8	4.2	26.4	0.0	3
ft-16a	65.8	5.8	25.2	0.0	3

Figure 1. Typical input data for **Petplot.** The ft-values are ID's, the four decimal values are component quantities (unnormalized), and the integer value represents a color code.

```
-115.0,   -24.2,   1
-115.0,   162.0,   1
 125.0,   -24.2,   1
 125.0,   162.0,   1
   0.0,    75.0    1
```

Figure 2. Scattered data file for use in ISM graphics program. The points represent maximum x and y values for sizing the illustration and an arbitrary z value that will prevent unwanted contour lines from appearing.

```
SETPEN 1,1,1
SETTXT 3,4,1.0,2
LGDFRM -1,-1,2
LGDBOX -9,3,30.0%,25.0%,0,0
LGDTXT 2," ","QUATERNARY DIAGRAM","TITLE AREA"
SETTXT 3,3,1.0,2
LGDTXT 2," "," "
LGDSYM 2,2,3,4,"Group 1"
LGDSYM 2,13,3,4,"Group 2"
LGDSYM 2,3,3,4,"Group 3"
SRFLNE 2,2,0
          81.65,   -14.18
           0.00,   160.82
SRFLNE 2,2,0
          81.65,   -14.18
        -111.54,    -5.19
           0.00,   160.82
(two SRFLNE commands omitted)
SRFTXT 2,0
           0.00,   163.82 "PHASE A"
(three SRFTXT commands omitted)
SETTXT 1,2,0.5,3
SRFSYM 2, 2,1,0
          -0.66,   155.58
SRFSYM 2, 2,1,0
           8.96,   124.44
```

Figure 3. Annotation file for use with the ISM graphics program. The values will allow the user to produce a finished illustration of the tetrahedral (or ternary) outline and the data points. SETPEN is an ISM command for selecting desired pen weights; SETTXT is an ISM command that determines text sizes and characteristics, LGDFRM is an ISM command that establishes a reference for the legend and causes a border to be drawn; other LGD commands cause the title and legend to be produced; SRFLN is an ISM command to draw lines, in this case the tetrahedron outline; SRFSYM is an ISM command to plot a symbol, here representing the data points; SRFTXT is an ISM command that causes text to be printed on the illustration at a specific location. The actual text string may be modified within this file using a text-editor.

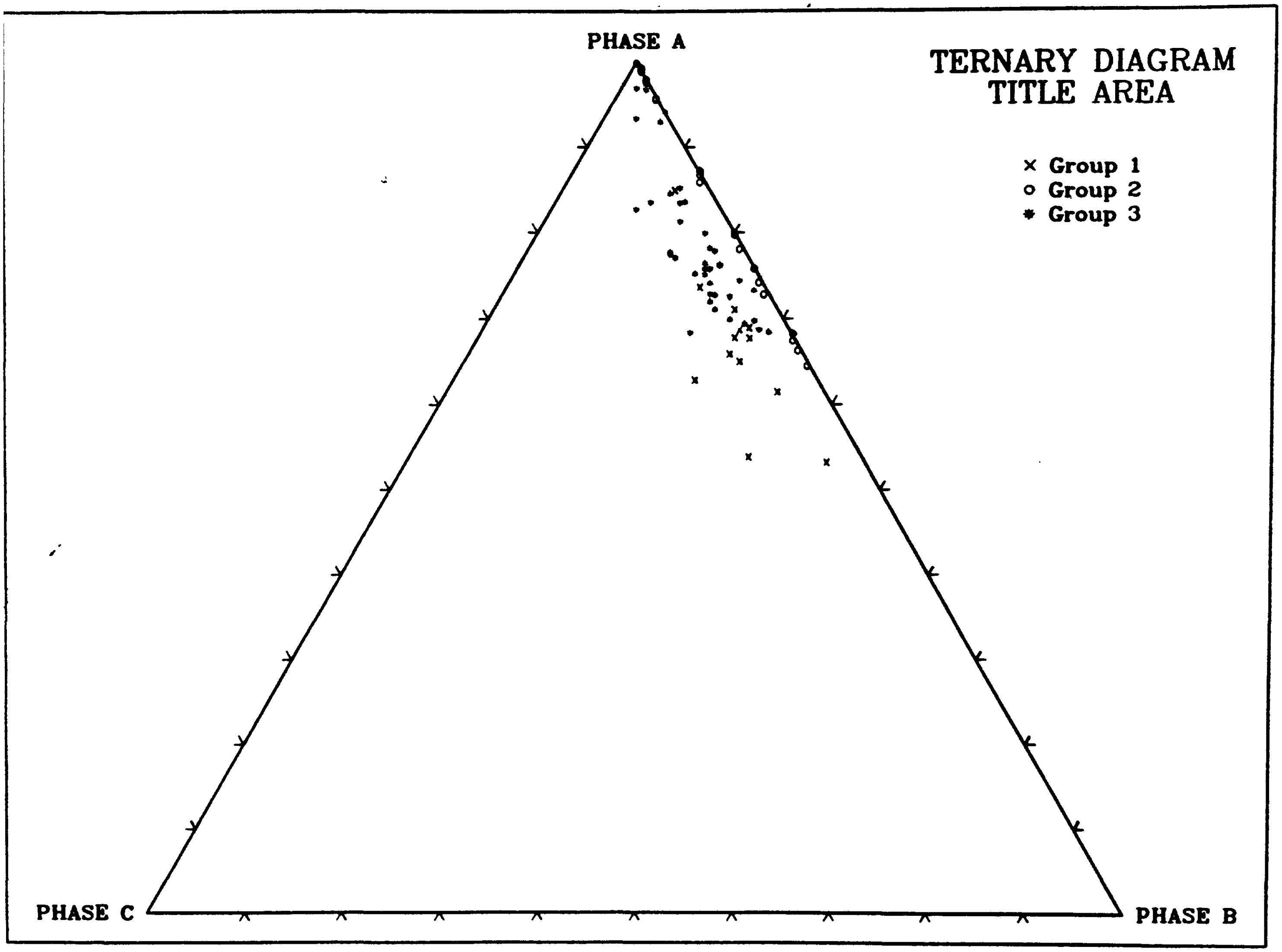

Figure 4. Ternary diagram of petrographic data showing generalized labels and 10% tic marks. The text string in these labels may be modified in the ISM annotation file using a text-editor.

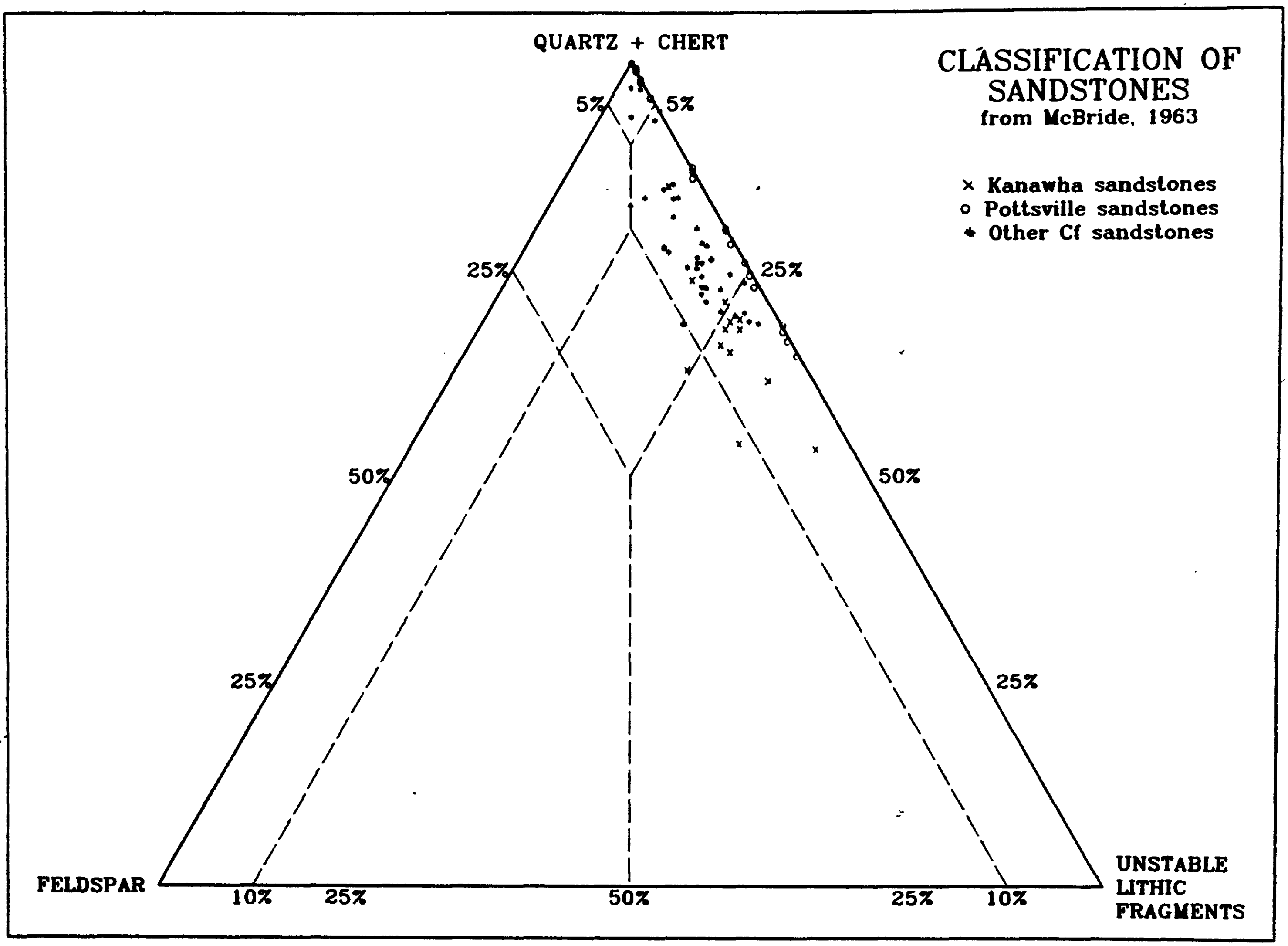

Figure 5. Ternary diagram of petrographic data showing the McBride, 1963, sandstone classification as an overlay on the data.

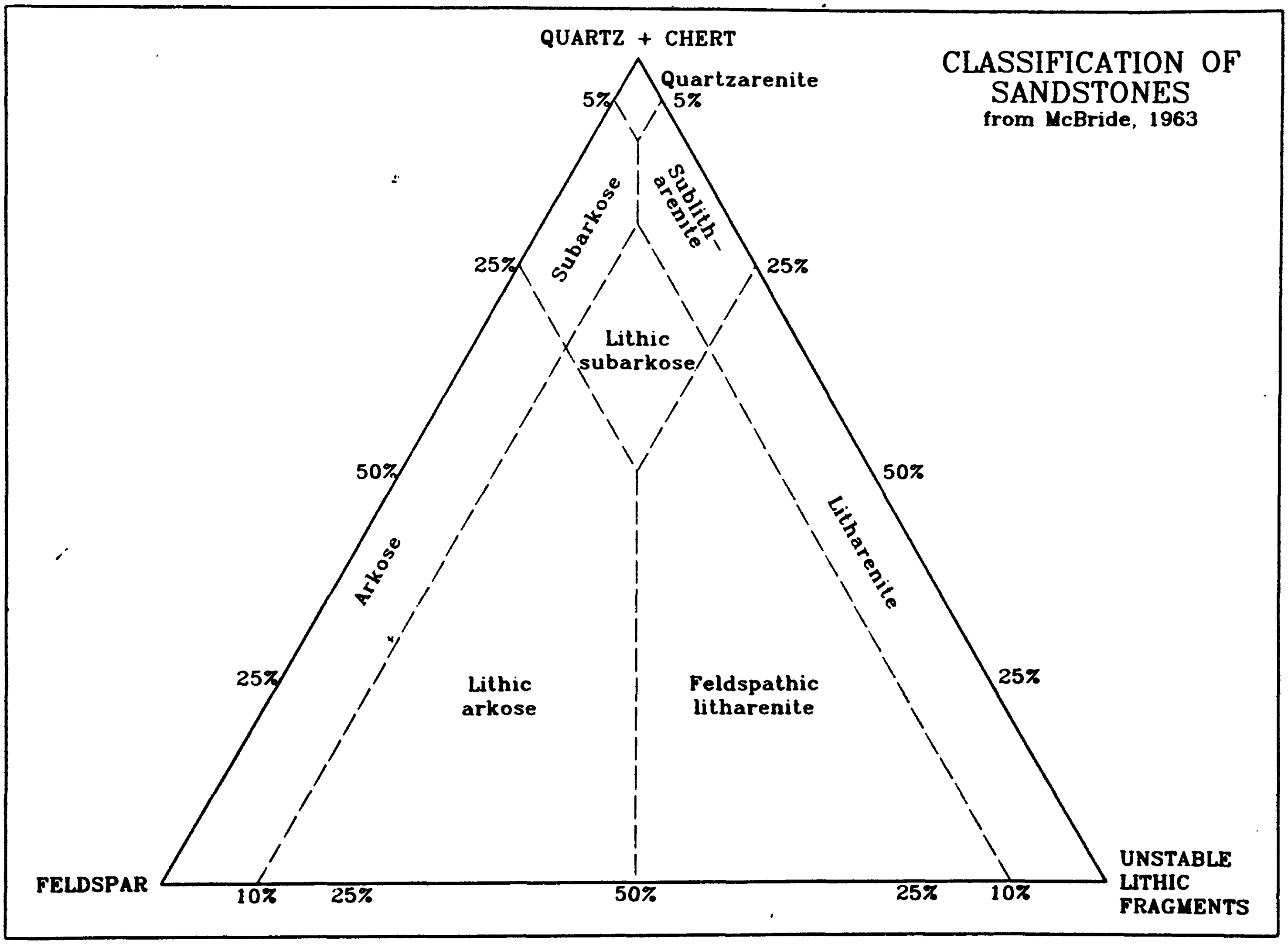

Figure 6. Reference ternary diagram showing the McBride, 1963, sandstone classification with the fields labeled but without data points.

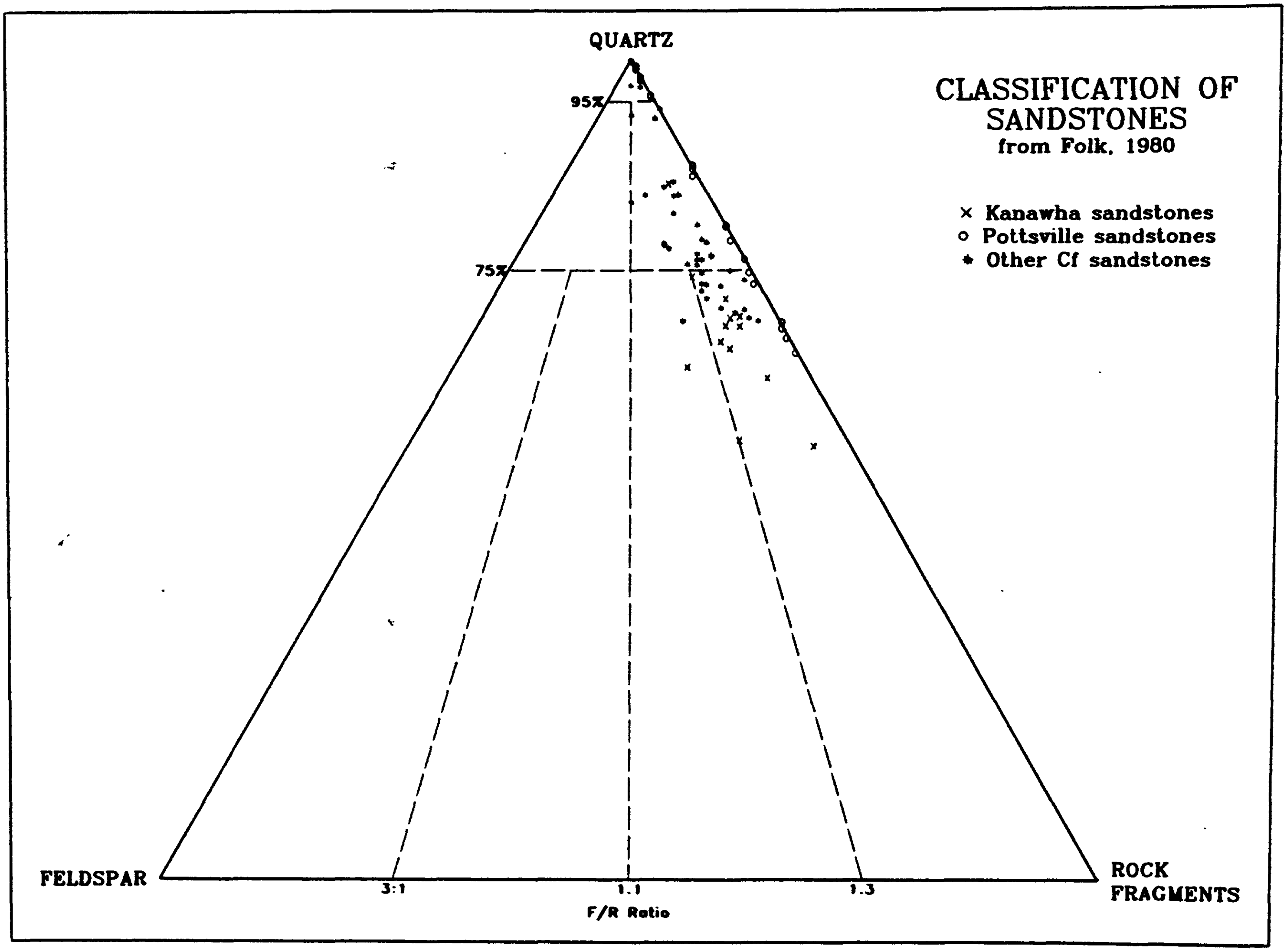

Figure 7. Ternary diagram of petrographic data showing the Folk, 1980, sandstone classification as an overlay on the data.

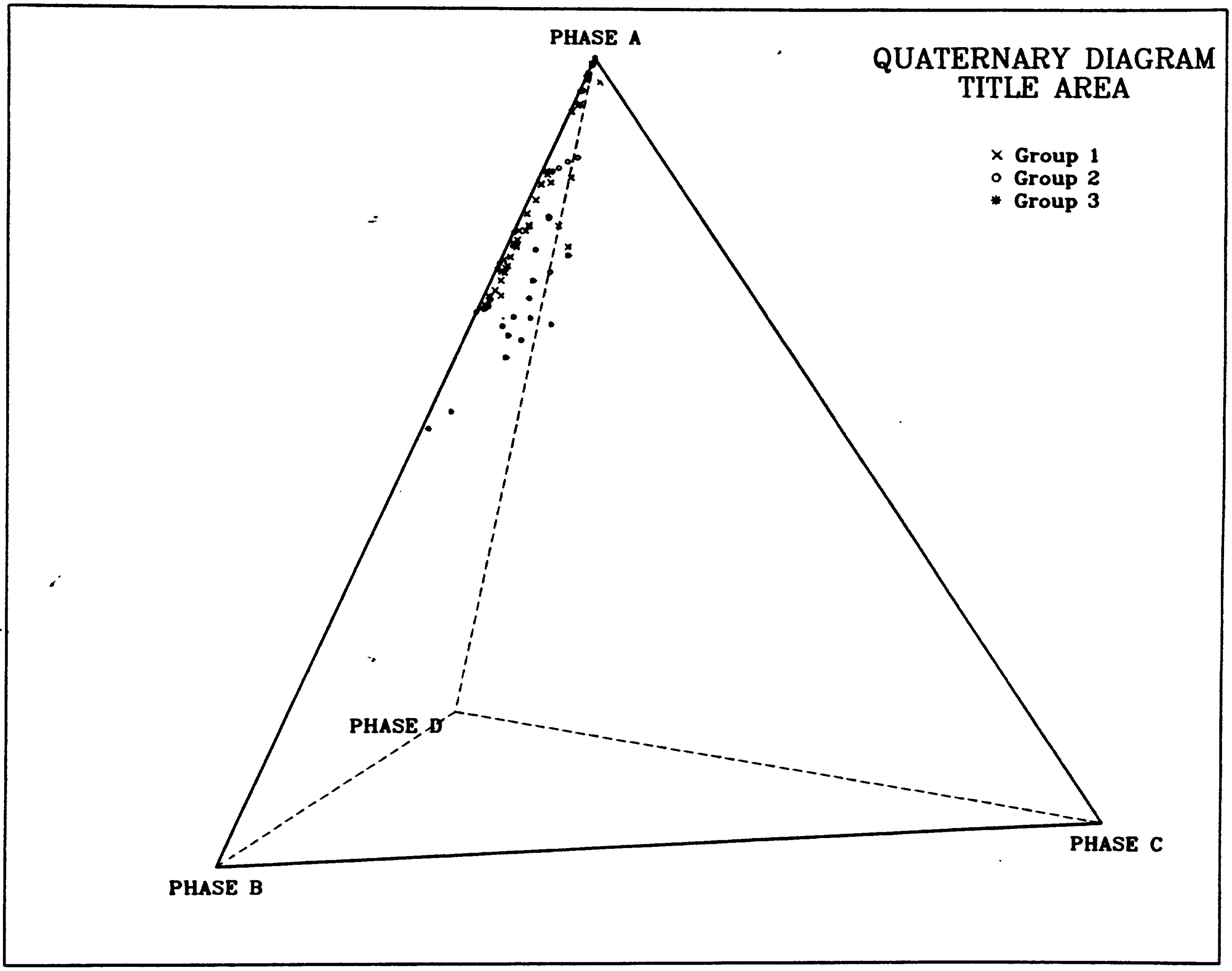

Figure 8. Quaternary diagram of petrographic data using a -10 degree x-axis rotation and a +45 degree y-axis rotation. The text string in the labels may be modified in the ISM annotation file using a text editor.

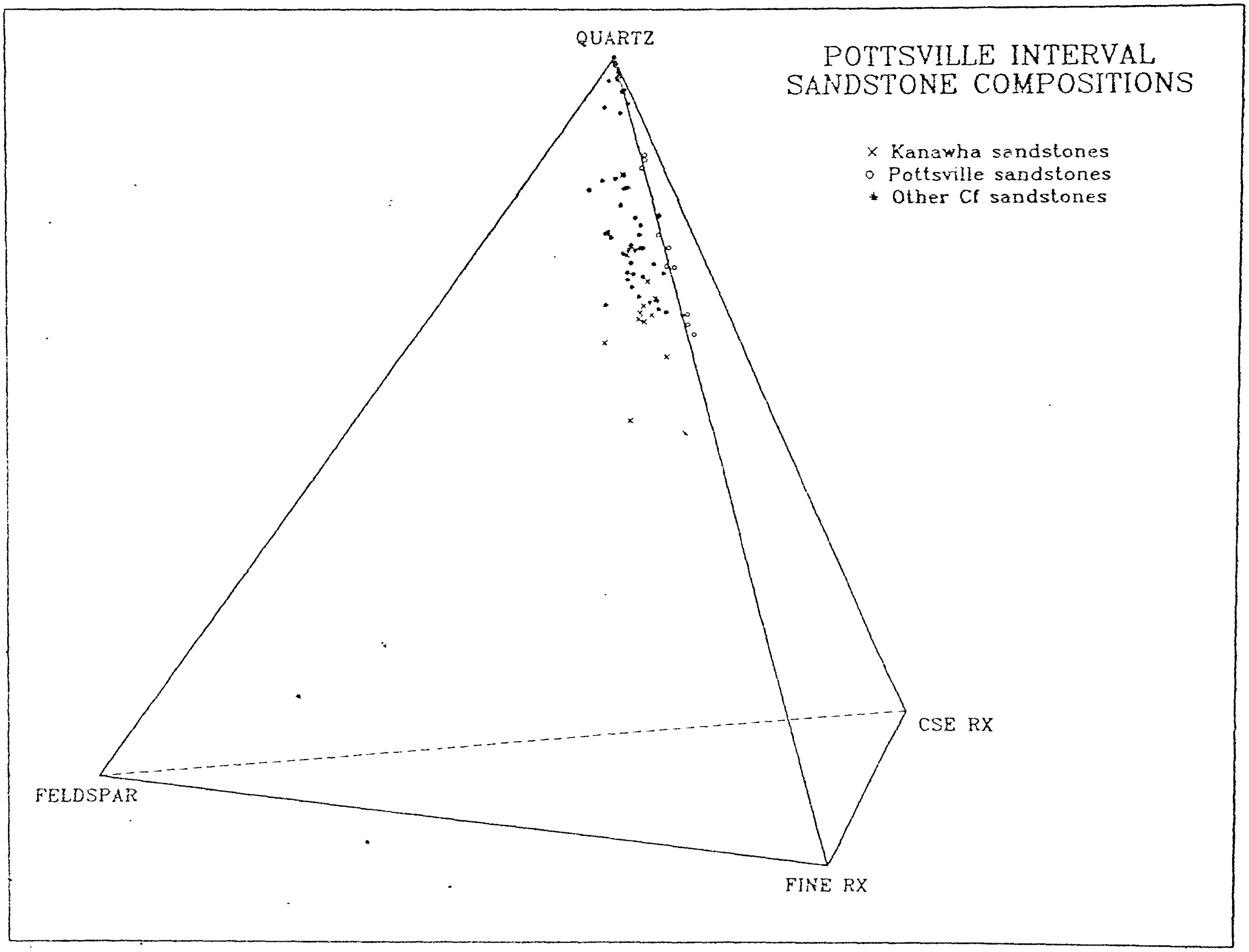

Figure 9. Quaternary diagram of petrographic data using a -10 degree x-axis rotation and a -45 degree y-axis rotation. The text string in the labels has been modified by editing the ISM annotation file using a text editor.