

OCTAVE ~ Another Scilab concept**Feb 15, 2008**

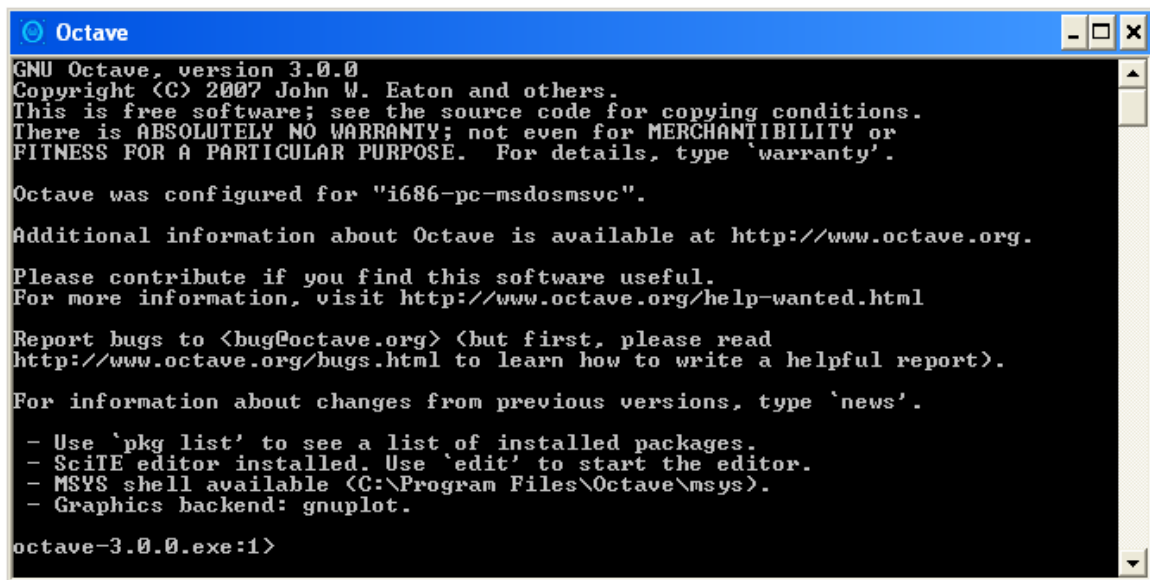
Download and install Octave from here.

<http://www.gnu.org/software/octave/>

The program is downloaded, and installed. The install process is very simple. OCTAVE is somewhat like MATLAB, which is part of the community of mathematical packages such as Scilab and Euler, for which I have provided textual as well as graphical dialling programs.

Octave is available at:- <http://www.gnu.org/software/octave/> and click the DOWNLOAD link and then scroll down the DOWNLOAD page and look for WINDOWS. It has a link for example to "The OCTAVE FORGE", click on the Octave Forge link and look down that page for the WINDOWS INSTALLER, look on that page for LATEST and that will have beneath it **octave-3.0.0-setup.exe** or something like that.. Click that latest install file and it installs and you are ready.

After installation double click on the OCTAVE desktop shortcut. And wait. Up comes a boring DOS like window, never fear, after it does its thing type "edit" (without the quotes). From then on edit, mess with the code, save, and in the original Octave window, enter the saved file's name. Of course, files must be saved in the path used by Octave. And file names are case sensitive.



```
GNU Octave, version 3.0.0
Copyright (C) 2007 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.

Octave was configured for "i686-pc-msdosmsvc".

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html

Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).

For information about changes from previous versions, type 'news'.

- Use 'pkg list' to see a list of installed packages.
- SciTE editor installed. Use 'edit' to start the editor.
- MSYS shell available (C:\Program Files\Octave\msys).
- Graphics backend: gnuplot.

octave-3.0.0.exe:1>
```

As you can see, Octave has a very boring DOS looking window, however, looks are deceiving. Type the word "**edit**" and a very context sensitive editor appears.

NOTE: Tested on Windows XP SP2.

NOTE: Octave is provided under the same licensing scheme as Euler and Scilab. It is free and donations are solicited.

Here is the Octave program for the h-dial.

TEXT/TABULAR VERSION

```
##
## hdial.m      A horizontal dial with longitude correction
##

## OCTAVE is somewhat like MATLAB, which is part of the community of mathematical
## packages such as Scilab and Euler, for which I have provided textual as well
## as graphical dialling programs.
##
## Octave is available at:-      http://www.gnu.org/software/octave/
##      and click the DOWNLOAD link
##      scroll down the DOWNLOAD page and look for WINDOWS
##      and it has a link for example to "The OCTAVE FORGE"
##      click on the Octave Forge link and
##      look down that page for the WINDOWS INSTALLER
##      look on that page for LATEST and that will have beneath it
##      octave-3.0.0-setup.exe      or something like that.
##      click that latest install file and it installs and you are ready.
##      after the usual virus checking, double click it.
##      Installation is very simple, and after installation
##      double click on the OCTAVE desktop shortcut. And wait.
##      Up comes a boring DOS like window, never fear, after it does its thing
##      type "edit" (without the quotes).
##
##      From then on edit, mess with the code, save, and in the original Octave
##      window, enter the saved file's name. Of course, files must be saved in
##      the path used by Octave.

clc;
printf ("\n\nwww.illustratingshadows.com\n\n");

## This code written by Simon Wheaton-Smith  FRI, MBCS, CITP  February 13, 2008

##sLat = input ("Lat: [32.75]:","s");
##printf ("Latitude is: %s\n", sLat);
fLat = input ("Lat: [32.75]:  ");

fLng = input ("Lng: [108.2]:  ");

fRef = input ("Ref: [105]:  ");

printf ("\n=====\\n");

printf ("Latitude is:      %5.2f\\n", fLat);
printf ("Longitude is:      %6.2f\\n", fLng);
printf ("Ref.Long is:        %6.2f\\n", fRef);
corh = (fLng - fRef)/15;
corm = corh *60;
printf ("Correction:  hh.h:  %5.2f      mm.m %2.1f \\n", corh, corm);

printf ("\n=====\\n");

hr = 6;
while (hr <19)
    hlat = sin(2*3.14162*fLat/360) * tan(2*3.1416*((hr-corh)*15)/360) ;
    hla = atan(hlat);
    hlad = hla * 360 / (2*3.14162);
    tod = (hr);
    printf ("Hour:   %3.0f   Hour Line Angle: %5.2f \\n", tod, hlad)
    hr++;
endwhile

printf ("\n=====\\n");
printf ("*** END ***\\n\\n");

## END
```

GRAPHICAL VERSIONS OF DIAL CODE

```
##
## hdialGR.m      A horizontal dial with longitude correction
##

clc;                                ## clear the log
printf ("\n\nwww.illustratingshadows.com\n\n");
printf ("Ignore the missing font message that may appear later.\n\n");

## This code written by Simon Wheaton-Smith  FRI, MBCS, CITP  February 13, 2008

##sLat = input ("Lat: [32.75]:","s");
##printf ("Latitude is: %s\n", sLat);
fLat = input ("Lat: [32.75]: ");

fLng = input ("Lng: [108.2]: ");

fRef = input ("Ref: [105]: ");

printf ("\n=====\\n");

printf ("Latitude is:      %5.2f\\n", fLat);
printf ("Longitude is:     %6.2f\\n", fLng);
printf ("Ref.Long is:      %6.2f\\n", fRef);
corh = (fLng - fRef)/15;
corm = corh *60;
printf ("Correction:  hh.h:  %5.2f      mm.m %2.1f \\n", corh, corm);

printf ("\n=====\\n");

## define a graphics area which we call:- gr      ## page 212 etc of manual
gr = axes();                                ## get a plotting area
## axis ("equal");                          ## force x to equal y
clf();                                      ## clear the current figure
hold ("on");                               ## each drawing adds to previous drawings
set (gca (), "defaultlinecolor", "black");
axis ([0,20,-10,10],"square");             ## force x to equal y
##                                           ## placed after the set GCA, it works

## dial center is defined as 10,0 and size limit is 10
xc=10;
yc=0;
xysize=10;

## do the loop for the desired hours
hr = 5;
while (hr <20)
    hlat = sin(2*3.14162*fLat/360) * tan(2*3.1416*((hr-corh)*15)/360) ;
    hla = atan(hlat);
    hlad = hla * 360 / (2*3.14162);
    tod = (hr);
    printf ("Hour:  %3.0f  Hour Line Angle: %5.2f \\n", tod, hlad)

    if (hlad < -45 && hr < 12)
        ## normal early morning hours
        x = -10;
        y = x / hlat;
        xx = [ xc, xc+x ];
        yy = [ yc, yc+y ];
        line (xx,yy);
    elseif (hlad >45 && hr < 12)
        ## non normal hours more than 90 degrees of noon
        x = -10;
        y = x / hlat;
        xx = [ xc, xc+x ];
        yy = [ yc, yc+y ];
        line (xx,yy);
    elseif (hlad > 45 && hr > 12)
        ## normal late afternoon hours
        x = 10;
        y = x / hlat;
        xx = [ xc, xc+x ];
        yy = [ yc, yc+y ];
        line (xx,yy);
    elseif (hlad < -45 && hr > 12)
        ## non normal late afternoon hours
        x = 10;
        y = x / hlat;
```

```

        xx = [ xc, xc+x ];
        yy = [ yc, yc+y ];
        line (xx,yy);
    else
        ## middle of day hours
        y = 10;
        x = y * hlat;
        xx = [ xc, xc+x ];
        yy = [ yc, yc+y ];
        line (xx,yy);
    endif
    th = num2str(hr);
    text ( xc+x, yc+y, th);
    tempangle = ceil(hlad*10);
    ta = num2str(tempangle/10);
    text ( xc+x, yc+y-0.7, ta);
    hr++;
endwhile

## display the graphics
set (gca (), "defaultlinecolor", "blue");
y = 10;
x = y;
xx = [ 5, 10 ];
yy = [ -10, -5 ];
line (xx,yy);
xx = [ 10, 15 ];
yy = [ -5, -10 ];
line (xx,yy);

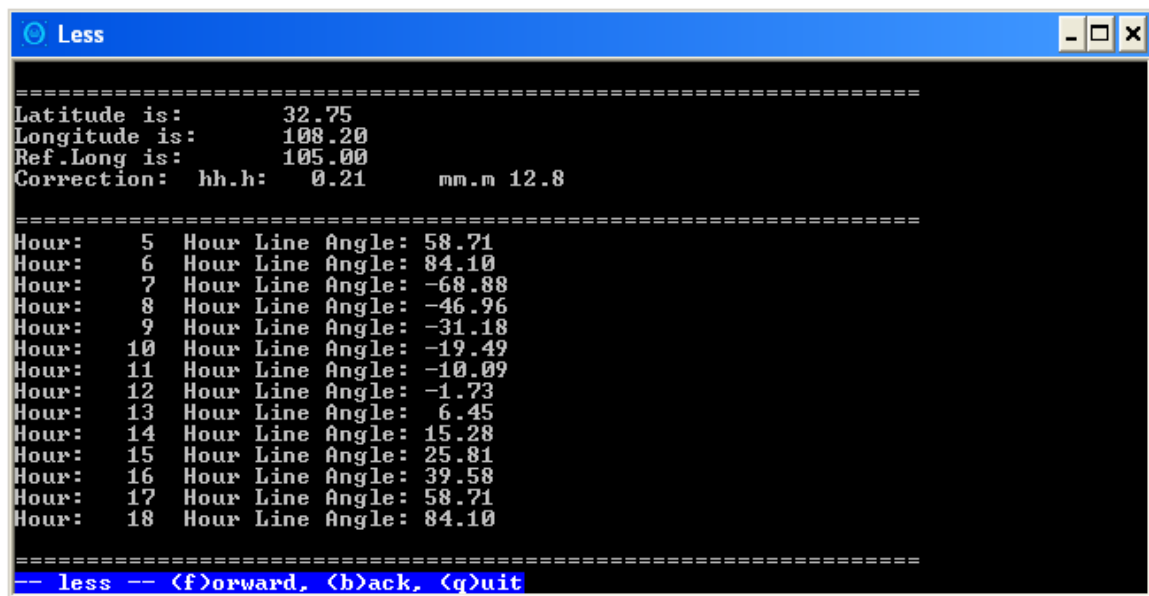
text ( 8,-6, "CAUTION: to verify the aspect ratio");
text ( 8,-7, "          use the 90 degree intersection");
text ( 8,-8, "          of the two 45 degree lines");
text ( 8,-9, "          to adjust aspect ratio.");
drawnow();

printf ("\n=====\\n");
printf ("*** END ***\\n\\n");

## END

```

When you enter "**hdialGR**" in the Octave DOS looking window, without the quotes, the program runs and displays two things. One is the detailed text display of hours and data.

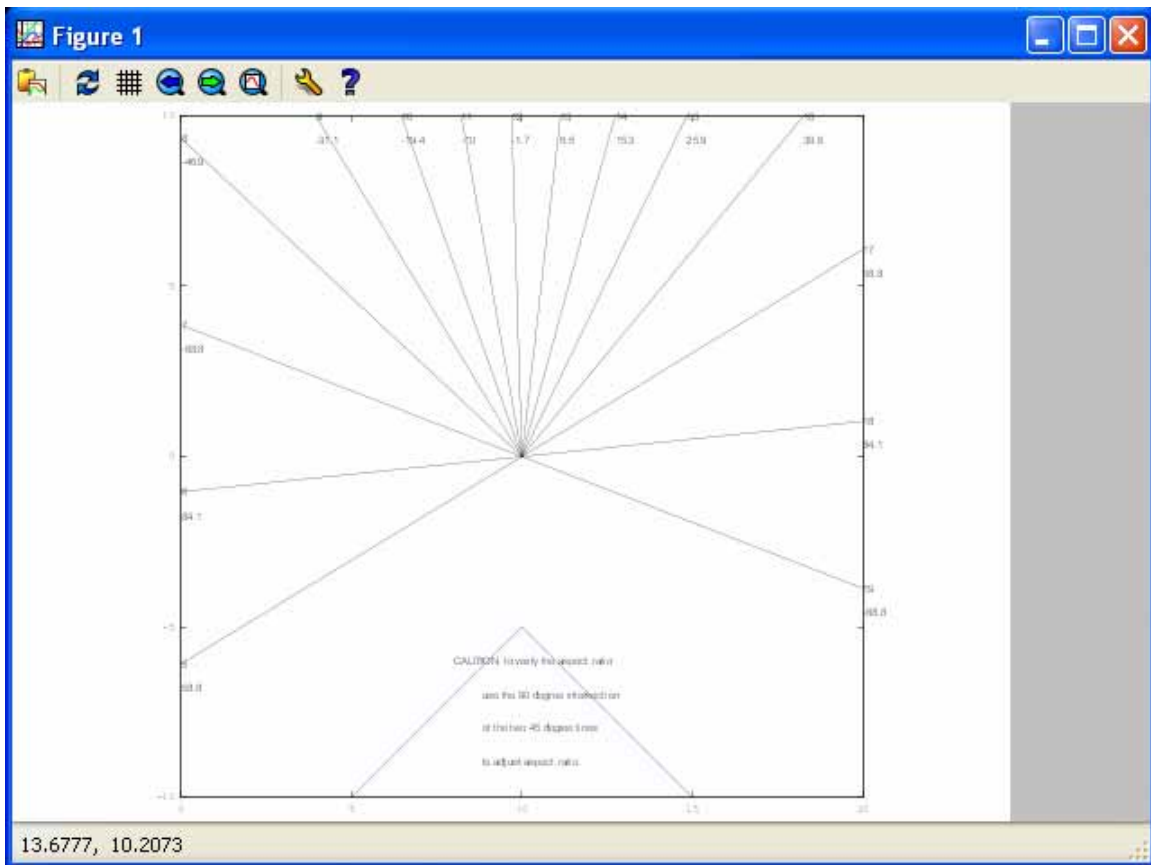


```

=====
Latitude is:      32.75
Longitude is:     108.20
Ref.Long is:      105.00
Correction:  hh.h:  0.21      mm.m 12.8
=====
Hour:   5  Hour Line Angle: 58.71
Hour:   6  Hour Line Angle: 84.10
Hour:   7  Hour Line Angle: -68.88
Hour:   8  Hour Line Angle: -46.96
Hour:   9  Hour Line Angle: -31.18
Hour:  10  Hour Line Angle: -19.49
Hour:  11  Hour Line Angle: -10.09
Hour:  12  Hour Line Angle: -1.73
Hour:  13  Hour Line Angle:  6.45
Hour:  14  Hour Line Angle: 15.28
Hour:  15  Hour Line Angle: 25.81
Hour:  16  Hour Line Angle: 39.58
Hour:  17  Hour Line Angle: 58.71
Hour:  18  Hour Line Angle: 84.10
=====
-- less -- <f>orward, <b>ack, <q>uit

```

The other display is the graphics display.



```

##
## vdialGR.m      A vertical dial with longitude correction
##

clc;                                ## clear the log
printf ("\n\nwww.illustratingshadows.com\n\n");
printf ("Ignore the missing font message that may appear later.\n\n");

## This code written by Simon Wheaton-Smith  FRI, MBCS, CITP  February 14, 2008

##sLat = input ("Lat: [32.75]:","s");
##printf ("Latitude is: %s\n", sLat);
fLat = input ("Lat: [32.75]: ");

fLng = input ("Lng: [108.2]: ");

fRef = input ("Ref: [105]: ");

printf ("\n===== \n");

printf ("Latitude is:      %5.2f\n", fLat);
printf ("Longitude is:     %6.2f\n", fLng);
printf ("Ref.Long is:      %6.2f\n", fRef);
corh = (fLng - fRef)/15;
corm = corh *60;
printf ("Correction:  hh.h:  %5.2f      mm.m %2.1f \n", corh, corm);

printf ("\n===== \n");

## define a graphics area          ## page 212 etc of manual
gr = axes();                       ## get a plotting area
clf();                             ## clear the current figure
hold ("on");                       ## each drawing adds to previous drawings
set (gca (), "defaultlinecolor", "black");
axis ([0,20,-10,10],"square"); ## force x to equal y (do after set gca)

## dial center is defined as 10,0 and size limit is 10
xc=10;
yc=0;
xysize=10;

## do the loop for the desired hours
hr = 5;
while (hr <20)
    hlat = cos(2*3.14162*fLat/360) * tan(2*3.1416*((hr-corh)*15)/360) ;
    hla = atan(hlat);
    hlad = hla * 360 / (2*3.14162);
    tod = (hr);
    printf ("Hour:  %3.0f  Hour Line Angle: %5.2f \n", tod, hlad)

    flag = 1;                      ## say line is displayable
    if (hlad < -45 && hr < 12)
        ## normal early morning hours
        x = -10;
        y = x / hlat;
        xx = [ xc, xc+x ];
        yy = [ yc, yc-y ];
        line (xx,yy);
    elseif (hlad >45 && hr < 12)
        ## non normal hours more than 90 degrees of noon
        x = -10;
        y = x / hlat;
        xx = [ xc, xc+x ];
        yy = [ yc, yc-y ];
        ## no line because can never display it  ## line (xx,yy);
        flag = 0;                  ## say line is not displayable
    elseif (hlad > 45 && hr > 12)
        ## normal late afternoon hours
        x = 10;
        y = x / hlat;
        xx = [ xc, xc+x ];
        yy = [ yc, yc-y ];
        line (xx,yy);
    elseif (hlad < -45 && hr > 12)
        ## non normal late afternoon hours
        x = 10;
        y = x / hlat;
        xx = [ xc, xc+x ];
        yy = [ yc, yc-y ];
        ## no line because can never display it  ## line (xx,yy);

```

```

        flag = 0;                ## say line is not displayable
    else
        ## middle of day hours
        y = 10;
        x = y * hlat;
        xx = [ xc, xc+x ];
        yy = [ yc, yc-y ];
        line (xx,yy);
    endif

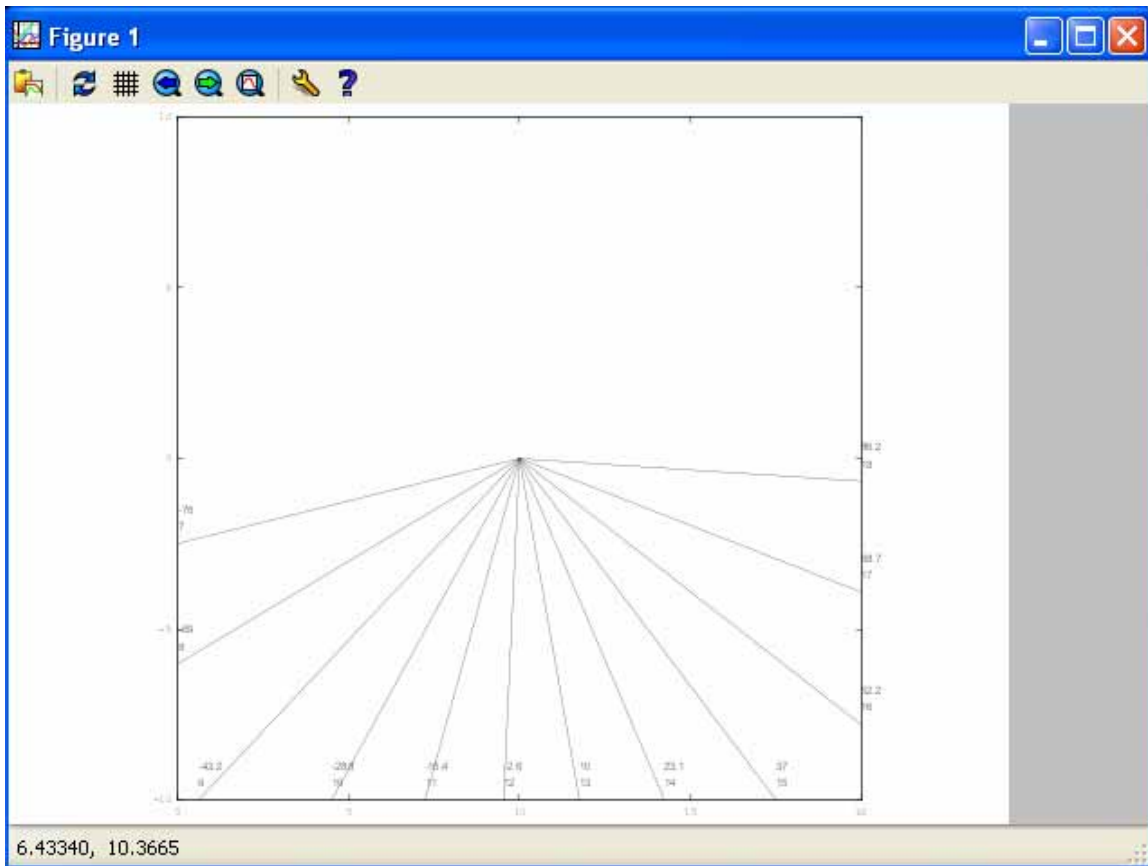
    if ( flag == 1)
        ## dont talk of lines that cant be displayed
        th = num2str(hr);
        text ( xc+x, yc-y+0.5, th);
        tempangle = ceil(hlad*10);
        ta = num2str(tempangle/10);
        text ( xc+x, yc-y+0.9, ta);
    endif
    hr++;
endwhile

## display the graphics
drawnow();

printf ("\n=====\\n");
printf ("*** END ***\\n\\n");

## END

```



The dial does not show hour lines that would be above the horizontal.

NOTES ON CONTROLLING GRAPHICS

The graphics area is defined first and the sequence of commands is important:-

```
## define a graphics area                                ## page 212 etc of manual
gr = axes();                                              ## get a plotting area
clf();                                                    ## clear the current figure
hold ("on");                                              ## each drawing adds to priors
set (gca (), "defaultlinecolor", "black");              ## force x equal y (after set gca)
axis ([0,20,-10,10],"square");
```

After the graphics area is defined, you may turn on or off the axes and their scales. Those scales have numbers obviously, and they can confuse reading the hours and the hour line angles. The following code will drop or keep Octave's axes and scales.

```
## let Octave do a box and scale (0 the default) or do our own (1)
if ( dbx == 1 )
    o = gca();
    set (o, "visible", "off");
endif
## get current graphical object
## dont draw scales
```

However, if you drop Octave's box (axes) and their scales, then the dial plate looks somewhat naked. So, you can add a box just before the graphics is displayed.

```
## display the graphics
if ( dbx == 1 )
    ## let Octave do a box and scale (0) or do our own (1)
    xx = [ 0, 20 ];
    yy = [ -10, -10 ];
    line (xx,yy);
    xx = [ 0, 20 ];
    yy = [ 2, 2 ];
    line (xx,yy);
    xx = [ 0, 0 ];
    yy = [ -10, 2 ];
    line (xx,yy);
    xx = [ 20, 20 ];
    yy = [ -10, 2 ];
    line (xx,yy);
endif
drawnow();
```

NOTE: The box drawn is the same for the vdialGR.m and vdecGR.m programs, but for the hdialGR.m program, a different box looks better.

```
if ( dbx == 1 )
    ## let Octave do a box and scale (0) or do our own (1)
    xx = [ 0, 20 ];
    yy = [ -10, -10 ];
    line (xx,yy);
    xx = [ 0, 20 ];
    yy = [ 10, 10 ];
    line (xx,yy);
    xx = [ 0, 0 ];
    yy = [ -10, 10 ];
    line (xx,yy);
    xx = [ 20, 20 ];
    yy = [ -10, 10 ];
    line (xx,yy);
endif
drawnow();
```