

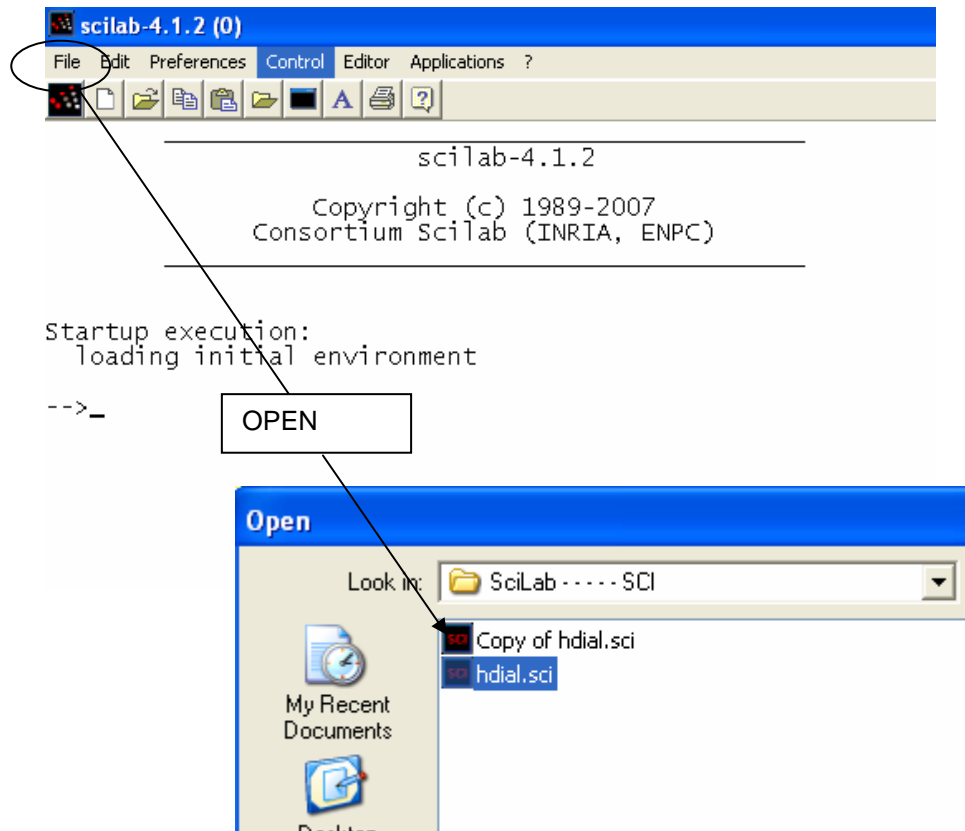
SCILAB NOTES**SciLab 4.1.2****Jan 21, 2008 1300**

NOTE: SciLab has been tested on Windows XP SP1 and on SP2. After installation, the computer must be restarted.

Home page: <http://www.scilab.org/>

Online manuals: <http://www.scilab.org/product/man/>

From the Scilab home page, download and install the latest Scilab. This used version 4.1.2. Then start up Scilab, and at startup first do a FILE, CHANGE DIRECTORY, and locate the folder with the desired program **hdial.sci**. Then do a FILE, OPEN and load it. In the editor click EXECUTE. Alternatively, instead of opening the program, once the CHANGE DIRECTORY has been done, enter "exec hdial.sci". You can also run the **vdial.sci** and **vdec.sci** programs.



Startup execution:
loading initial environment

-->_

OPEN

FILE, then OPEN, and locate the folder and the hdial.sci program. This brings up the editor, and the program can be changed, and executed then and there. It can also be run with the exec command.

```

1 // Horizontal Dial
2
3 // Get latitude data first
4 lat=32.75
5 lat = input ("Latitude: ")
6 latR = lat * 2 * 3.1416 / 360
7 tlatR = tan(latR)
8 slatR = sin(latR)
9
10 // Get longitude and its correction
11 lng=108.2
12 lng = input ("Longitude: ")
13 ref=105
14 ref = input ("Ref.Longitude: ")

```

Either way, answer the prompts for the latitude, longitude, and reference longitude.

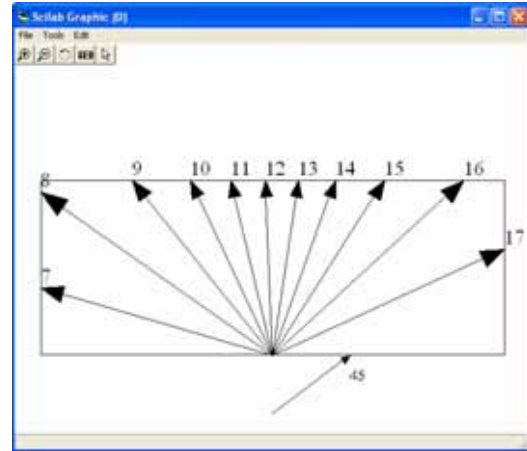
If you write your own programs, and get the "more y or n" message, ensure you add the ";" to all program statements, that eliminates echoing of values set by statements.

CAUTION:

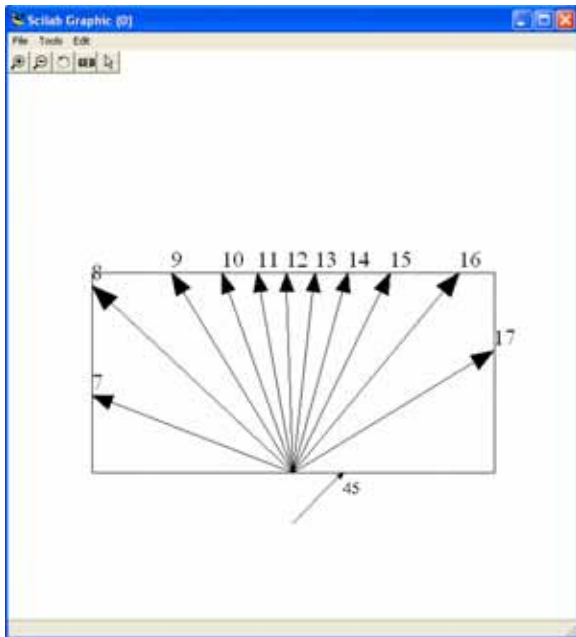
The graphical depiction may have the wrong aspect ratio, however a 45 degree line is shown so you can rescale it. The program solution is to use the "square" command, which the hdial.sci, vdial.sci and vdec.sci programs do use.

With no SQUARE command →

With the SQUARE command ↓



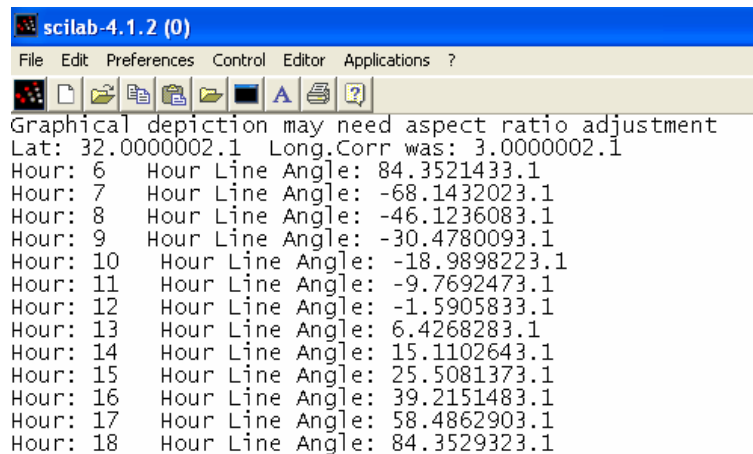
wrong aspect ratio



good aspect ratio

CAUTION: Scilab will crash if the program is run a couple of times, simply reload Scilab before each use of these Scilab programs.

Either way, the program ends with a table of hours and hour line angles.



```

scilab-4.1.2 (0)
File Edit Preferences Control Editor Applications ?
Graphical depiction may need aspect ratio adjustment
Lat: 32.0000002.1 Long.Corr was: 3.0000002.1
Hour: 6 Hour Line Angle: 84.3521433.1
Hour: 7 Hour Line Angle: -68.1432023.1
Hour: 8 Hour Line Angle: -46.1236083.1
Hour: 9 Hour Line Angle: -30.4780093.1
Hour: 10 Hour Line Angle: -18.9898223.1
Hour: 11 Hour Line Angle: -9.7692473.1
Hour: 12 Hour Line Angle: -1.5905833.1
Hour: 13 Hour Line Angle: 6.4268283.1
Hour: 14 Hour Line Angle: 15.1102643.1
Hour: 15 Hour Line Angle: 25.5081373.1
Hour: 16 Hour Line Angle: 39.2151483.1
Hour: 17 Hour Line Angle: 58.4862903.1
Hour: 18 Hour Line Angle: 84.3529323.1
    
```

Scilab has a lot in common with C, FORTRAN, and other procedural languages. Like many of the newer languages, the documentation is excellent if you already know the language. However there are no obviously locatable sample programs to show you a working program that gets user input, processes it, and generates simple output. These programs do that, and simply. These programs do not have all the code that is in the DeltaCAD examples, that code corrects trigonometric quirks relating to quadrants to display data. The purpose of this code is to introduce the Scilab language. The h-dial program is shown below. The source for the hdial, vdial, and vdec dials is in the zip file.

```

// Horizontal Dial ~ ~ ~ For a given latitude and longitude

// Get latitude, longitude and reference longitude
lat = input ("Latitude [eg 32.75]: ");
latR = lat * 2 * 3.1416 / 360;
tlatR = tan(latR);
slatR = sin(latR);

lng = input ("Longitude [eg 108.2]: ");
ref = input ("Ref.Longitude [eg 105]: ");
clng = lng - ref;

xset ("font", 1, 8);

// Set the parameters for the graphics area
xc = 0.65; // x of dial center ) xc=0.65
yc = 0.1; // y of dial center ) yc = 0.1
ysize = .8; // size of the y scale ) ysize=0.8
xsize = ysize; // size of the x scale ) xsize = ysize
// This corrects for aspect ratio - must go hand in hand with about values)
square ( -.2, -.2, 1.5, 1.5 ); // xmin ymin xmax ymax )

for hr = 7:17; // 6:19 is normal hour range
    // However extra code is needed for angles such as for 0600 when dial is
    // west of reference longitude, and for angles such as 1800 when the dial
    // is east of longitude. The purpose of this program was to show the use
    // of Scilab's language and techniques.
    // Simon Wheaton-Smith FRI, MBCS, CITP
    // www.illustratingshadows.com Jan 11, 2008 0815

    // calculate the local hour angle in radians L H A = local hour angle of the sun
    lha = (15 * hr) - clng; // local hour angle = hour * 15 degrees
    lhaR = lha * 2 * 3.1416 / 360; // in radians
    tlhaR = tan(lhaR); // tlhaR is tan of hour angle

    // calculate the hour line angle per se H L A = hour line angle on the plate
    // hrLnAng = atan ( sin(lat) * tan (lha) )
    hlaR = atan(slatR * tlhaR); // hour line angle in radians
    hlaD = ( hlaR * 360 ) / ( 2 * 3.1416 ); // hour line angle in degrees
    
```

```
// determine the end point for early, midday, or late hours
if ( hlaD < -45 ) then;
    // before 0900
    // assuming x as a fixed end point, what is the y value for the hour
    xe = xc - xsize ; // x end = 1 x size plus dial center x
    ye = yc - (xsize / tan(hlaR)); // y end = x size * formula
elseif ( hlaD > 45 ) then;
    // after 1500
    // assuming x as a fixed end point, what is the y value for the hour
    xe = xc + xsize ; // x end = 1 x size plus dial center x
    ye = yc + (xsize / tan(hlaR)) ; // y end = x size * formula
else // 0900 to 1500
    // assuming y as a fixed end point, what is the x value for the hour
    ye = yc + ysize ; // y end = 1 y size plus dial center y
    xe = xc + (ysize * tan(hlaR)) ; // x end = y size * formula
end;

// draw an hour line
xa = [ xc xe ];
ya = [ yc ye ];
xarrows(xa, ya);

// name the hour line
// xstring( 1, sin(lat)*tan(lha) , "hr" )
xset ("font", 1, 8);
xstring( xe , ye , string(hr), 0) ; // show hour at end of hour line
xset ("font", 1, 2);
xstring( xe , ye+0.1 , string(int(hlaD)), 0);

end;

// draw a pretty box
xrect ( xc-xsize, yc+ysize, 2*xsize, ysize) ; // upper left point, then sizes

// draw a calibration line they must scale to 45 because Scilab aspect ratios
// leave a lot to be desired.
xa = [ xc xc+0.2 ] ; // center a 45 degree line and use
ya = [ yc-0.2 yc ] ; // x for both x and y to force a 45
xset ("font", 1, 4);
xstring( xc+0.2 , yc-0.1 , string(45), 0) ; // used for aspect ratio
xarrows(xa, ya);

// state the hour line angles now
printf ("Graphical depiction may or may need aspect ratio adjustment\n");
printf ("Lat: %3.1f Long.corr was: %3.1f \n", lat, clng);
for hr = 6:18 ; // print the hour and hour line angles
    printf ("Hour: %i Hour Line Angle: %3.1f \n", hr, ( (atan(slatR * (tan(((15 * hr) -
        clng) * 2 * 3.1416 / 360)))) * 360) / (2 * 3.1416)));
end;
printf ("*** ENTER *** to end program\n");
// END
```

As a reminder, the Scilab coordinate system is as follows, with 0,0 bottom left, however boxes start with the top left coordinate. The following program and depiction shows a problem with aspect ratios, unless corrected. The program fix is to use the "square" command, which is done in the h-dial.sci program.

```

1 // Scilab coordinates
2
3 // draw an hour line
4 xa = [ 0 0.75 ]
5 ya = [ 0 0.75 ]
6 xarrows(xa, ya)
7
8 xstring( 0, 0, " 0, 0", 0)
9 xstring( 0, 0.75, " 0 0.75", 0)
10 xstring( 0.75, 0, " 0.75 0", 0)
11 xstring( 0.75, 0.75, " 0.75 0.75", 0)
12
13 xrect ( 0, 1, 1, 1) // upper left point, then sizes
14
15
16

```

Without the SQUARE command

