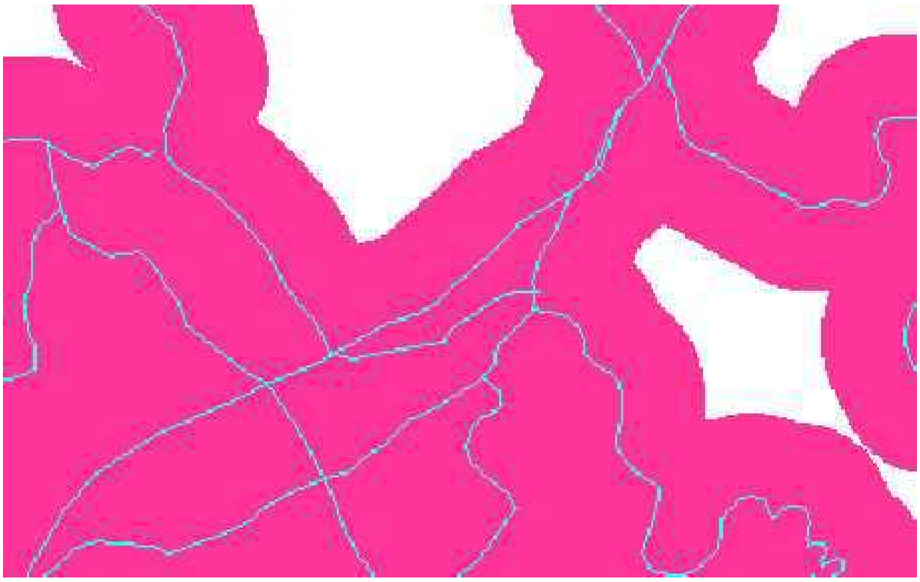
Distance and Raster Maps

There are two parts to this exercise
making a buffer using **`r.buffer`** and
finding the "downhill" path on a DEM using **`r.drain`**.

Making buffers on a road, river, or other linear feature

Let's say that you don't want to build a campground anywhere near a highway. First you must decide how far you need to be, say 1 kilometer. r.buffer will determine all of the cells that are within this area, like this.



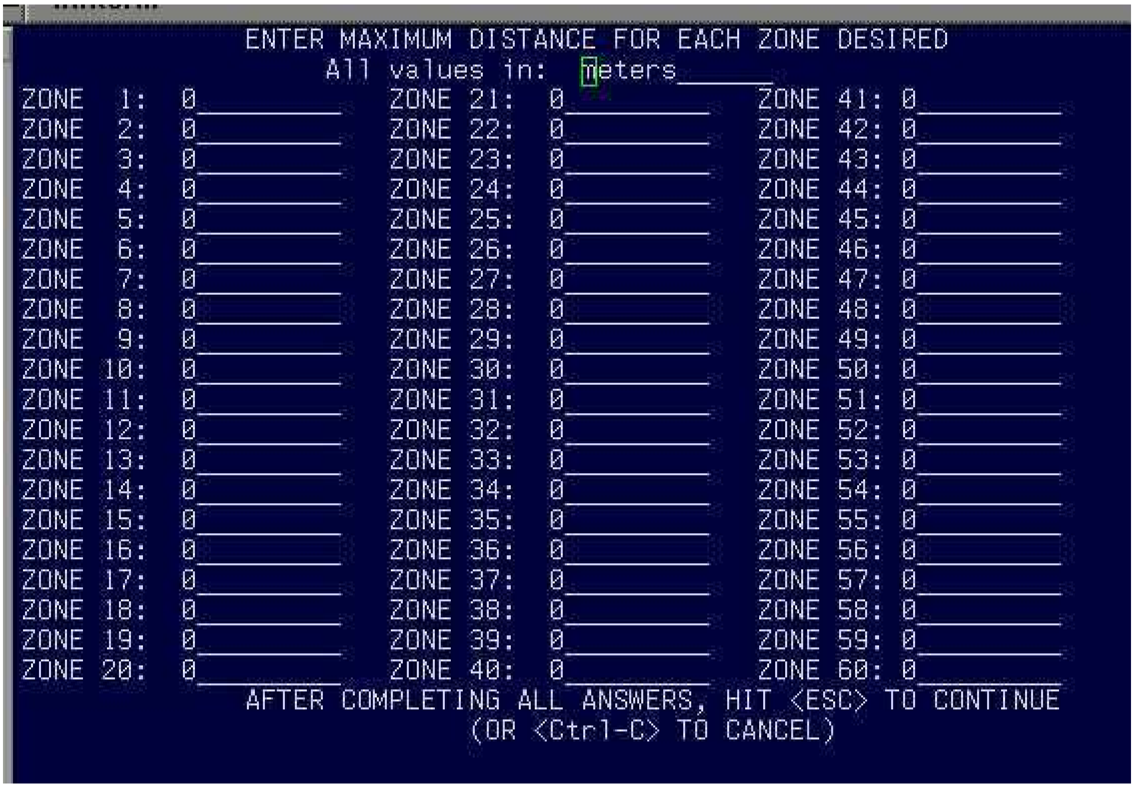
The pink areas are within 1 km of the grey lines (the roads) and the white areas are now deemed suitable for the campground. In GRASS, the map to be buffered must be a raster map and you can have up to 60 different buffer zones around the linear featur e.

Make your own road buffer map and one showing the distance to the streams.

Log on to gilbert and start the GIS and a monitor (don't remember how? go to week 1)

Making a buffer around roads

enter
`r.buffer`
and follow the instructions for the input file name (use the raster layer "roads")
and give the name "road.buffer" or something similar to the output map.
You will be greeted by the following screen



type return once to advance to zone 1 and
type in the MAXIMUM value for the buffer (use "**500**" for example)
then press the escape key when you're done.
NOTE: If you make a mistake, keep hitting the return key until you come back around
ANOTHER NOTE: If your screen isn't sized large enough, this screen may not work correctly.
The shell should tell you the progress and return to the normal GRASS prompt when it's done. When it is done type
`d.rast road.buffer`
to see the results.

If you hate "screens" this can be done via command-line ala
`r.buffer input=roads output=road.buffer distances=500`

Making multiple buffers

Now let's make a map showing the distance to the streams in the demo area. First, let's look at the are using the satellite image. Type
`d.rast false.color`
and overlay the streams by typing
`d.rast streams -o` (the minus 'oh' overlays only non-zero values from the streams map)
These are only the major streams on the map.

Now make a multiple-buffer map called "dist.to.stream" by running **`r.buffer`** again, but this time entering 200, 400, 600, etc up to 3000 meters for zone 1, 2, 3 etc up to 15. Then press escape to run the buffering program . In specific
run **`r.buffer`**
use "streams" for the input map
use "dist.to.stream" for the output map
and the buffers as above.
Then display your handiwork by typing
`d.rast dist.to.stream`

Please delete the **road.buffer map** when you're done by typing

`g.remove`
and following the instructions (or **`g.remove rast=road.buffer`** if you know what you're doing).

Finding a "least cost" path along a surface

As explained in the text in Chapter 4, it is sometimes necessary to find the distance between a point and another via a "surface" of least "cost." One example of this is a raindrop running across a topographic surface. We can do thi s in GRASS using the r.drain command as follows.

To orient yourself, display the topography using
`d.rast dem`
Now we need to locate a point near the summit of the mountain in the top center of the image (that's th red colors). Type
`d.where`
click on a point and write down the x (east: should be near 600,000) and y (north: should be near 4 million) coordinates for use in a minute (you don't need to copy anything after the decimal; I think meter precision will do, false as it is). **Quit d.where** by clicking the right button on the graphics screen (for Macs that is the right arrow key instead). Make sure that the graphics screen is the forward panel before you do so.
Now find out where that point "drains" to by using the command
`r.drain`
For the "cost elevation" surface, use the Digital Elevation Model layer "dem"
then enter an output name, like "drain.example"
and the previously defined coordinates separated by a comma like this
`632004,4524679`
It will ask for a second coordinate pair after you confirm the first by saying
`enter option>`
which you should ignore, just hit enter and it will be done.

Now display the "drain" map over the dem by typing
`d.rast drain.example -o`

The topgraphic model for this area is not very good so you may not drain all the way off the map (there are lots of "pits" created by linear data errors).

Please **erase** the **drain.example map** when you're done with it.

When you're done, go back the the Tutorial Page

HELP: send e-mail to harbor