Building a 2D vel model with Vel2Grid using polygons

Alberto M. López Venegas, Ph.D.

Welcome to the wonderful world of polygons, vertexes and edges. It is real simple once you get used to it and find the trick, because it is tricky.

I have attached my Vel2Grid section of my nonlinloc control file. The important thing to do is to make a sketch of what you want to do: If its along the X axis or the Y axis and the depths
of interest. The idea is to make simple polygons that will resemble a downgoing slab. The rest
can be solved by having a background 1D layered velocity model (as you will see in the LAYER
statements at the end of the vel2grid section). To start off, you will need to change VGRID statement
to VELOCITY so that when you plot it, you can see how the velocities of your newly created velocity model look like. Once you are done testing, and you know your velocity model is what you wanted
you will need to change this to SLOWNESS or something else for earthquake relocations.
Also the cpt (GMT color palette table) should be in accord with your velocities, so that you can see the variation in velocities, say from 3 km/sec to 10 km/sec. If your range is large, then all of your polygons will have the same color.

I did a simple sketch of the NorthAmerican slab subducting beneath the Caribbean plate. I went from 17 degreess North to 21 degrees North (if I recall well) and then changed those to kilometers.
I did my profile North-South as I was interested in the subduction north of Puerto Rico, so this is important
to visualize. If you need a profile that is not parallel to the main axis, then you will need to do some
more complicated things and make transformations. You can draw your intended velocity model as a function of depth in your sketch. Estimate what layers
and polygons you will need. Estimate their dimensions and velocities. Then, after everything is drawn
you will have to translate all of that into vertexes and edges. These two will make out the polygons.
As you can see in the section below I first put the vertexes; these are points with latitudes (in my case, these
may change for you if doing as a function of longitude or as an off-axis transformation) as a function
of depth. Each vertex will be identified by an index, then its depth, then its location in X-axis, then Y-location
(all in km). Notice that Y-location is always the same: in my case this was the mid-point in longitude: What I end up having is a subducting slab to the south that propagates to each side longitudinally (hence a 2D velocity model).
Once you have identified all of your necessary vertexes, try to see which ones are entirely necessary. I say this because I began placing everything as polygons, but then as it evolved, i did just

a couple of them and let the horizontal LAYER command to do the rest: basically, we have horizontal
layers interrupted by a slab going down. Now that you have identified your polygons: sediments, crust,
upper mantle, slab oceanic crust, slab lithosphere, accretionary wedge, mantle, etc.... you need to specify
edges. Edges will rely on the vertex id number you have just done above. I would go in clockwise order.
By trial and error polygons should be simple, don' t do complicated things because otherwise the program
will confuse and output something that you don' t want. So ideally 4 to 6 vertexes have worked for me, and simple forms...you' ll see if you do my case below. The downgoing slab was subdivided in four segments at the end, and it began with only one. I did 4 segments of approximately 10 km thickness for slab crust and immediately below them four thicker segments for the upper mantle. Then the polygon statements. These should be specified with the edges on the second line, and once again identify them clockwise. Specify how many edges comprise the polygon, then declare its velocities, densities if you care, and their gradients. It is important to declare the 2DTO3DTRANS statement before any VERTEX, EDGE or POLYGON
statement. This statement declares where the origin is (x-origin , y-origin, and the rotation). This turned out
to be tricky. I did it several times with roation =0 or 90 with alternating Y-loc and X-loc for my points in the VERTEX section. At the end, the best way to do this is to leave the rotation to 90 if you want your profile
to be extended along the x-axis (along latitude variation) and have vertex y-direction fixed. I know it is odd but it works. The most important thing to do is once you are done computing it, is to plot it so that you know everything is how you planned it to be. You should use Anthony' s Grid2GMT command for plotting (http://alomax.free.fr/nlloc/soft6.00/Grid2GMT.html):
Vertical cross section (V):

Grid2GMT InputControlFile GridRoot Outroot V PlotType iX1 iY1 iX2 iY2

I hope this works and let me know if you hit a wall.
Saludos y mucha suerte y éxito,

Alberto

```
#
=============================================================================
#
=============================================================================
# Vel2Grid control file statements
#
=============================================================================
#
#

# output filename root
```

```
# (VGOUT <output file root>)
# Layer 2DGrid
VGOUT  ./model/layer

# wave type
# (VGTYPE wave_type (P, S))
VGTYPE P
VGTYPE S

# grid description
# (GRID  num_grid_x  num_grid_y  num_grid_z
#       orig_grid_x  orig_grid_y  orig_grid_z
#       d_grid_x d_grid_y d_grid_z
#       type
#    (float) num_grid_x/y/z : number of nodes along x/y/z axis
#    (float)    orig_grid_x : x location of grid origin (0,0,0) in km
pos east
#    (float)    orig_grid_y : y location of grid origin (0,0,0) in km
pos north
#    (float)    orig_grid_z : z location of grid origin (0,0,0) in km
pos down
#    (float)   d_grid_x/y/x : grid spacing along  x/y/z axis
#    (char[])  type : (VELOCITY = km/s, VELOCITY_METERS = m/s,
#               SLOWNESS = s/km,
#               VEL2 = vel**2,
#               SLOW2 = slow**2, SLOW_2_METERS = slow**2 ((s/m)**2),
#               SLOW_LEN = slow*d_grid)
#
#
# Layer 2DGrid (NOTE: num_grid_x must be = 2 for 2D grids)
# VGGRID  90 70 175 0.0 0.0 -1.0 5.0 5.0 1.0  VELOCITY
#AJL#VGGRID   105 70 175 0.0 0.0 -1.0 5.0 5.0 5.0  SLOW_LEN
# VGGRID 10 7 17 -1.0 0.0 0.0 50.0 50.0 50.0  SLOW_LEN
#AJL#
#VGGRID  420 2000 420   0.0 0.0 -1.0   0.5 0.5 0.5  VELOCITY
VGGRID  420 500 500   0.0 0.0 -1.0   1 1 1  SLOW_LEN

#
-----------------------------------------------------------------------------
# velocity model description
#
-----------------------------------------------------------------------------

#
-----------------------------------------------------------------------------
#2DTO3DTRANS 250 250  90 # POLYGON SEction Test: three polygons witha
gently dipping slope
#VERTEX 1 0 0 0 #VERTEX 2 0 500 0
#VERTEX 3 180 0  0
#VERTEX 4 320 0  0
#VERTEX 5 140 500 0
#VERTEX 6 400 0  0
#VERTEX 7 400 500 0
#EDGE 12 1 2
#EDGE 13 1 3
```

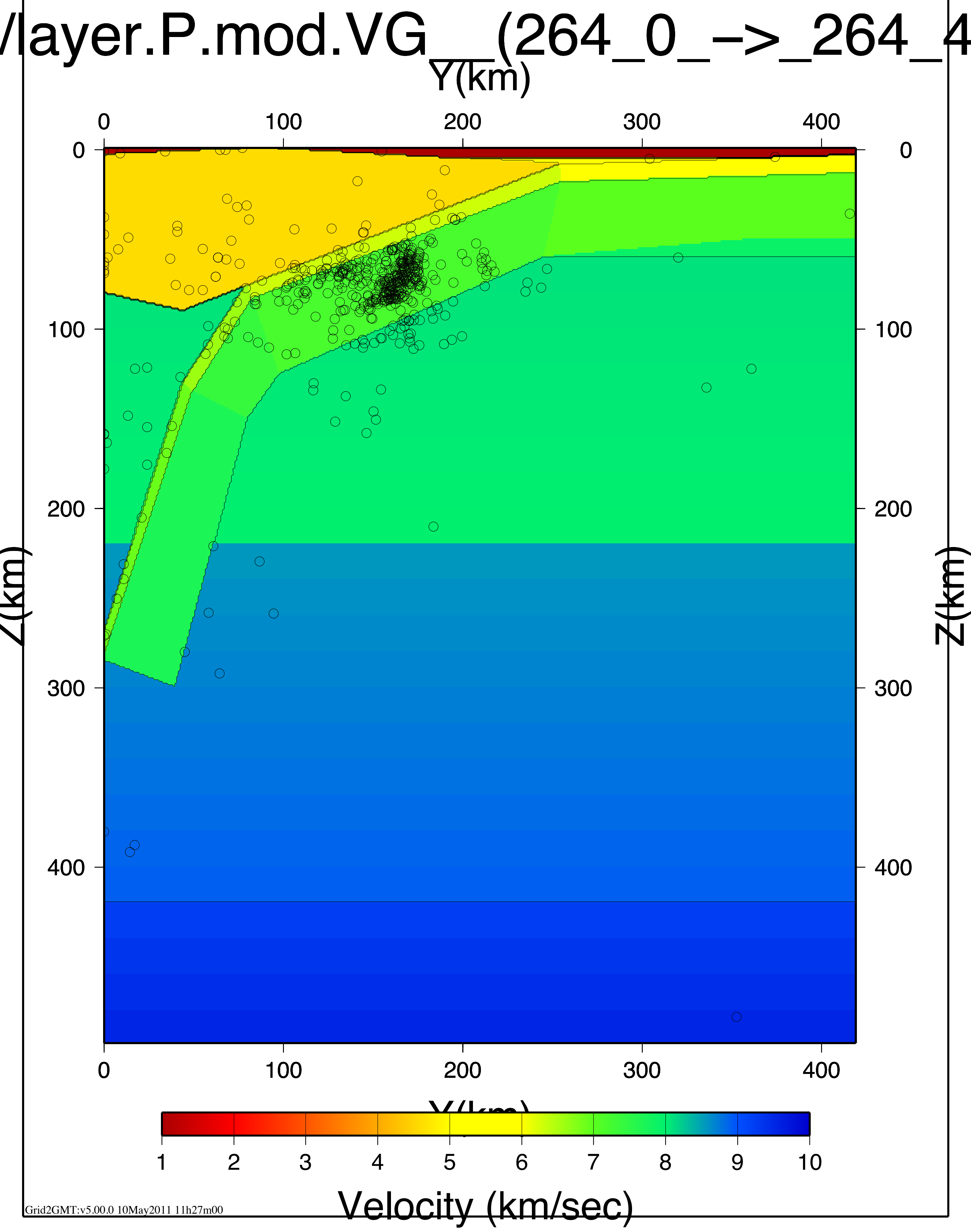
```
#EDGE 14 1 4
#EDGE 32 3 2
#EDGE 43 4 3
#EDGE 45 4 5
#EDGE 46 4 6
#EDGE 67 6 7
#EDGE 57 5 7
#EDGE 25 2 5
#POLYGON2 id_num n_edges depth Vp_top Vp_grad Vs_top Vs_grad p_top
p_grad [NEW_LINE] edge1, edge2, ... #POLYGON2 111 3 0 5 0 2.7 0 2.7 0
#12 13 32
#POLYGON2 222 4 200 6 0 3 0 3.3 0
#32 25 45 43
#POLYGON2 333 4 350 8 0 4 0 4 0 #45 57 67 46
## #LAYER   0.0  1.0 0.00    2.67  0.00  2.7 0.0
#
#
#   POLYGON Section of south-dipping NA slab underneath CA
#2DTO3DTRANS
#xOrig (km)     yOrig (km)      rotation (deg)
2DTO3DTRANS 264 0 90
#VERTEX Velocity Model
#    id_num  zloc (km)     xloc (km)    yloc (km)
# xloc is fixed because is at the center of a grid from -67W to -62W
VERTEX 110 -1   88.456 264
VERTEX 111 8.3  254.31 264
VERTEX 112 90   44.228 264
VERTEX 113 80   0      264
VERTEX 114 2.8  0      264
VERTEX 115 75   80     264
VERTEX 221 3    420    264
VERTEX 222 10   420    264
VERTEX 223 20   254.31 264
VERTEX 224 100  88.456 264
VERTEX 225 150  44.228 264
VERTEX 226 300  0      264
VERTEX 227 270  0      264
VERTEX 228 130  44.228 264
VERTEX 229 90   88.456 264
VERTEX 331 120  0      264
VERTEX 551 45   420    264
VERTEX 552 60   245.31 264
VERTEX 553 125  98     264
VERTEX 554 150  80     264
VERTEX 555 300  40     264
VERTEX 661 300  420    264
VERTEX 556 285  0      264
VERTEX 557 135  50     264
VERTEX 771 83   83     264
VERTEX 991 18   255    264
VERTEX 992 13   420    264
#
#EDGE Velocity Model
#id_num vertex1 vertex2
EDGE 11 110 111
```

```
EDGE 12 111 112
EDGE 13 112 113
EDGE 14 113 114
EDGE 10 114 110
EDGE 20 111 221
EDGE 21 221 222
EDGE 22 222 223
EDGE 23 223 224
EDGE 24 224 225
EDGE 25 225 226
EDGE 26 226 227
EDGE 27 227 228
EDGE 28 228 229
EDGE 29 229 111
EDGE 30 331 113
EDGE 31 228 331
EDGE 32 112 228
EDGE 41 227 331
EDGE 42 228 557
EDGE 43 115 771
EDGE 44 111 991
EDGE 45 221 992
EDGE 51 226 555
EDGE 52 554 555
EDGE 53 553 554
EDGE 54 552 553
EDGE 55 552 551
EDGE 56 551 222
EDGE 57 556 557
EDGE 58 554 557
EDGE 59 227 556
EDGE 61 555 661
EDGE 62 661 555
EDGE 70 228 554
EDGE 71 553 115
EDGE 72 115 112
EDGE 73 115 228
EDGE 74 557 771
EDGE 80 552 111
EDGE 81 771 991
EDGE 82 553 771
EDGE 91 551 221
EDGE 92 991 992
EDGE 93 991 552
EDGE 94 551 992
EDGE 95 555 556
#POLYGON2 Velocity
Model
#id_num n_edges depth (km) Vp_top (km/s) Vp_grad (km/s) Vs_top (km/s)
Vs_grad (km/s) p_top (km/m**3) p_grad (km/m**3) Edges
POLYGON2 1 5 0 4.59 0 2.73 0 2.7 0
10 11 12 13 14
#POLYGON2 3 6 90 8.07 0 4.65 0 2.7 0
#13 30 31 28 29 12
#POLYGON2 3 4 90 8.07 0 4.65 0 2.7 0
```

```
#13 30 31 32
#POLYGON2 2 10 8.605 6.5 0 3.5 0 2.7 0
#20 29 28 27 26 25 24 23 22 21
#POLYGON2 4 3 130 8.04 0 4.45 0 2.7 0
#27 41 31
#POLYGON2 5 10 20 7.4 0 4.45 0 2.7 0
#22 23 24 25 51 52 53 54 55 56
POLYGON2 5 4 20 7.7 0 4.6 0 2.7 0
95 52 58 57
#POLYGON2 6 6 60 8.08 0 4.47 0 2.7 0
#52 53 54 55 62 61
POLYGON2 7 4 110 7.4 0 4.4 0 2.7 0
58 53 82 74
POLYGON2 8 4 80 7.2 0 4.3 0 2.7 0
82 54 93 81
POLYGON2 9 4 80 7.0 0 4.17 0 2.7 0
92 93 55 94
POLYGON2 10 4 80 6.0 0 3.6 0 2.7 0
20 44 92 45
POLYGON2 11 4 80 6.3 0 3.75 0 2.7 0
12 43 81 44
POLYGON2 12 4 80 6.6 0 3.9 0 2.7 0
73 42 74 43
POLYGON2 13 4 80 6.9 0 4.1 0 2.7 0
27 59 57 42
#
#
LAYER   0.0  1.0 0.00    2.67  0.00  2.7 0.0
LAYER   5.0   5.0      0.00  2.9   0.00  2.7 0.0
LAYER   15.0  7.0      0.00  4.2   0.00  2.7 0.0
LAYER   50.0  7.9      0.00  4.64  0.00  2.7 0.0
LAYER   60.0  8.088611 0.00  4.477886 0.00 2.7 0.0
LAYER   80.0  8.076248 0.00  4.470515 0.00 2.7 0.0
LAYER  100.0  8.063886 0.00  4.463144 0.00 2.7 0.0
LAYER  120.0  8.051523 0.00  4.455773 0.00 2.7 0.0
LAYER  140.0  8.039160 0.00  4.448402 0.00 2.7 0.0
LAYER  160.0  8.026797 0.00  4.441030 0.00 2.7 0.0
LAYER  180.0  8.014434 0.00  4.433659 0.00 2.7 0.0
LAYER  200.0  8.002071 0.00  4.426288 0.00 2.7 0.0
LAYER  220.0  7.989709 0.00  4.418917 0.00 2.7 0.0
LAYER  220.0  8.558951 0.00  4.643901 0.00 2.7 0.0
LAYER  240.0  8.597426 0.00  4.657900 0.00 2.7 0.0
LAYER  260.0  8.635903 0.00  4.671900 0.00 2.7 0.0
LAYER  280.0  8.674380 0.00  4.685900 0.00 2.7 0.0
LAYER  300.0  8.712857 0.00  4.699900 0.00 2.7 0.0
LAYER  320.0  8.751335 0.00  4.713900 0.00 2.7 0.0
LAYER  340.0  8.789812 0.00  4.727900 0.00 2.7 0.0
LAYER  360.0  8.828289 0.00  4.741900 0.00 2.7 0.0
LAYER  380.0  8.866766 0.00  4.755900 0.00 2.7 0.0
LAYER  400.0  8.905241 0.00  4.769899 0.00 2.7 0.0
LAYER  420.0  9.236308 0.00  4.990832 0.00 2.7 0.0
LAYER  440.0  9.338698 0.00  5.049176 0.00 2.7 0.0
LAYER  460.0  9.441089 0.00  5.107521 0.00 2.7 0.0
LAYER  480.0  9.543480 0.00  5.165865 0.00 2.7 0.0
LAYER  500.0  9.645871 0.00  5.224209 0.00 2.7 0.0
```

```
LAYER  520.0  9.748262 0.00  5.282554 0.00 2.7 0.0
LAYER  540.0  9.850653 0.00  5.340898 0.00 2.7 0.0
LAYER  560.0  9.953043 0.00  5.399242 0.00 2.7 0.0
LAYER  580.0 10.055434 0.00  5.457587 0.00 2.7 0.0
LAYER  600.0 10.157825 0.00  5.515931 0.00 2.7 0.0
LAYER  620.0 10.188736 0.00  5.531501 0.00 2.7 0.0
LAYER  640.0 10.219711 0.00  5.546985 0.00 2.7 0.0
LAYER  660.0 10.250687 0.00  5.562469 0.00 2.7 0.0
LAYER  670.0 10.266173 0.00  5.570210 0.00 2.7 0.0
LAYER  670.0 10.751321 0.00  5.945129 0.00 2.7 0.0
LAYER  680.0 10.782464 0.00  5.974363 0.00 2.7 0.0
LAYER  700.0 10.844736 0.00  6.032844 0.00 2.7 0.0
LAYER  720.0 10.906984 0.00  6.091336 0.00 2.7 0.0
LAYER  740.0 10.969206 0.00  6.149838 0.00 2.7 0.0
LAYER  760.0 11.031403 0.00  6.208350 0.00 2.7 0.0
LAYER  780.0 11.082099 0.00  6.246872 0.00 2.7 0.0
LAYER  800.0 11.118520 0.00  6.261171 0.00 2.7 0.0 #
#
=============================================================================
# END of Vel2Grid control file statements
#
=============================================================================
#
=============================================================================
```

Alberto M. López Venegas, Ph.D.
Puerto Rico Seismic Network
Department of Geology University of Puerto Rico - Mayagüez
P.O. Box 9017
Mayagüez, Puerto Rico 00681-9017
http://prsn.uprm.edu/~alberto


/layer.P.mod.VG__(264_0_–>_264_4
Y(km)
0
100
200
300
400
0
100
200
300
400
Z(km)
Z(km)
Y(km)
1
2
3
4
5
6
7
8
9
10
Velocity (km/sec)
Grid2GMT:v5.00.0 10May2011 11h27m00