

CoveMaps.R

The process of generating the bathymetric maps is performed by three R scripts (R is a computer language):

1. ExtractRegions.R - This script extracts the data sets specific for a small defined region of the lake.
2. CoveMaps.R - This script generates the actual map of a cove using the data generated by the script specified under (1)
3. BatchRun.R - This script runs all of the small defined regions and generates all of the maps (30 as of this writing). This is the only process that is somewhat time consuming on the computer.

This note describes the second of the two R scripts.

Appendix A has the full script that produces the bathymetry map for one of the coves.

There are several things that one should be aware of:

1. This is a volunteer effort. This did not cost anyone anything. This was all done on my personal time.
2. This is still a work in progress because there are a number of technical issues that need to be resolved to make the most accurate maps possible. A lot more attention has to be paid to the boundary conditions and other interpolation algorithms for the irregularly spaced measures need to be tried. Furthermore, some older data should be folded into the analysis.
3. The R language was being learned as the work was being done and hence one will come across ways of doing things with R that are more efficient and more elegant, and perhaps more accurate. Waiting for a process to finish on the computer was ever an issue. A lot of Google searches were performed and some of the most useful, or groundbreaking solutions, are put as comments in the script.

The maps that are on the website are produced entirely with R. Three data sets were used:

1. The contour of the lake
2. The bathymetry measurements
3. The boat slip locations

Each of these files comprise data for the whole lake. The ExtractRegion.R script extracts only those data that are pertinent to the cove or lake area under investigation. That cove or area is defined by a rectangular bounding box with the coordinates of the box and name of the area encoded in a small file.

PLV: 2/8/2013

Appendix A

R Script to Produce the bathymetric maps for Deep Creek Lake Coves.

```
4. # Title:          Contours using Akima interpolation and Loess smoothing
5. # Description:    Provide a nice looking plot of contours for a specified cove or lake section
6. # The bounding box is used to define a matrix of points within which the cove is located.
7. # Date:          10/9/2012
8. # Project:       Bathymetry of Deep Creek lake
9. # File:          CoveMapsV10x.R
10. # Author:        Pete Versteegen
11. # Revised:       11/15/12; this has been under development for a while; it's final now; some small fixings to work on
12. # Revised:       11/16/12; new inout data descriptor file
13. # Revised:       11/21/12; added a graphics descriptor file to specify graphics parameters unique to each cove
14. #               It's now pretty much final.
15. #=====
16. # Reference Material
17. # http://stackoverflow.com/questions/11666197/create-colour-grid-over-map-in-r
18. # http://stackoverflow.com/questions/10047870/plotting-interpolated-data-on-map/10058844#10058844
19. # http://stackoverflow.com/questions/8563334/average-values-of-a-point-dataset-to-a-grid-dataset/8563683#8563683
20. # http://www.slideshare.net/tjagger/r-spatial-analysis-using-sp
21. # http://hosho.ees.hokudai.ac.jp/~kubo/Rdoc/library/geoR/html/polygrid.html
22. # http://stackoverflow.com/questions/10047870/plotting-interpolated-data-on-map
23. # par( "usr" ) returns a vector with the extreme coordinates of the user's plotting region
24. # Ref: http://www.statmethods.net/advgraphs/axes.html
25. # Ref: http://sphaerula.com/legacy/R/placingTextInPlots.html
26. # Ref: http://stat.ethz.ch/R-manual/R-devel/library/graphics/html/text.html
27. # Ref: http://onertipaday.blogspot.com/2008/09/fitting-text-under-plot.html
28. # Reference for the following: http://www.image.ucar.edu/GSP/Software/Fields/Help/image.plot.html
29. # http://stat.ethz.ch/R-manual/R-patched/library/graphics/html/legend.html
30. #=====
31.
32. begTime <- Sys.time()
33.
34.
35. #=====
36. # Load the needed libraries - quit the output when loading, supposedly...
37.
38. suppressMessages(library(akima))
39. suppressMessages(library(maptools))
40. suppressMessages(library(PBSmapping))
41. suppressMessages(library(RColorBrewer))
42. suppressMessages(library(fields))
43. suppressMessages(library(sp))
44. suppressMessages(library(tgp))
45. suppressMessages(library(splancs))
46. suppressMessages(library(SDMTools))
47. suppressMessages(library(spatstat))
48. suppressMessages(library(raster))
49. suppressMessages(library(geoR))
50. suppressMessages(library(gtools))
51. suppressMessages(library(GISTools)) # North arrow and map scale
52. suppressMessages(library(lattice))
53. suppressMessages(library(rgeos)) # To manipulate polygons.
54. suppressMessages(library(calibrate)) # To label the points of interest
55.
56.
57. #=====
58. # Read the problem specification
59.
60. con <- file("case_input.txt", "rt")
61. project <- readLines(con,1) # Root name of the file
62. resolut <- readLines(con,2) # Resolution in feet
63. close(con)
```

```

64. resolution <- as.numeric(resolut)
65. project
66.
67. #=====
68. # Read the supplemental information for the cove
69.
70. fileName <- paste("./run_files/", project, "_runfile", "_", resolution, ".txt", sep="")
71. fileConn <- file(fileName, "r") # open an input file connection
72. project_data <- read.table(fileConn, header=FALSE, sep="\t", comment.char = "#")
73. project_data
74. close(fileConn)
75.
76. projectp <- trim(as.character(project_data[1,2]))
77. areaname <- trim(as.character(project_data[2,2]))
78. nobathpoints <- as.numeric(as.character(project_data[3,2]))
79. resolution <- as.numeric(as.character(project_data[4,2]))
80. nshore <- as.numeric(as.character(project_data[5,2]))
81. nslips <- as.numeric(as.character(project_data[6,2]))
82. xminbox <- as.numeric(as.character(project_data[7,2]))
83. xmaxbox <- as.numeric(as.character(project_data[8,2]))
84. yminbox <- as.numeric(as.character(project_data[9,2]))
85. ymaxbox <- as.numeric(as.character(project_data[10,2]))
86. areap <- as.numeric(as.character(project_data[11,2]))
87. milesp <- as.numeric(as.character(project_data[12,2]))
88. nx <- as.numeric(as.character(project_data[13,2]))
89. ny <- as.numeric(as.character(project_data[14,2]))
90. nx
91. ny
92. #=====
93. # Define the bounding box as a polygon
94.
95. xbox <- c(xminbox, xmaxbox, xmaxbox, xminbox)
96. ybox <- c(yminbox, yminbox, ymaxbox, ymaxbox)
97. box_cbind <- cbind(xbox, ybox)
98. deltax <- (xmaxbox-xminbox)/nx
99. deltay <- (ymaxbox-yminbox)/ny
100.
101.
102. #=====
103. # Read the cove outline
104.
105. fileName <- paste("./run_files/", project, "_shoreline.txt", sep="")
106. fileConn <- file(fileName, "r") # open an input file connection
107. coveName2 <- readLines(fileConn, n=1)
108. coveoutline <- read.table(fileConn, header=TRUE, sep="\t", comment.char = "#")
109. close(fileConn)
110.
111. xcove <- coveoutline$x
112. ycove <- coveoutline$y
113. ncove <- length(xcove)
114. cove_cbind <- cbind(xcove, ycove)
115.
116.
117. #=====
118. # Read the boat slip locations
119.
120. fileName <- paste("./run_files/", project, "_boatslips.txt", sep="")
121. fileConn <- file(fileName, "r") # open an input file connection
122. coveName3 <- readLines(fileConn, n=1)
123. docks <- read.table(fileConn, header=TRUE, sep=" ", comment.char = "#")
124. close(fileConn)
125.
126. xdock <- docks$x
127. ydock <- docks$y

```

```

128.ndocks <- length(xdock)
129.dock.cb <- cbind(xdock, ydock)
130.
131.
132.#=====
133.# Read the locations of points of interest
134.
135.fileName <- paste("./run_files/", project, "_poi.txt", sep="")
136.fileConn <- file(fileName, "r") # open an input file connection
137.covename4 <- readLines(fileConn, n=1)
138.pois <- read.table(fileConn, header=TRUE, sep=" ", comment.char = "#")
139.close(fileConn)
140.
141.poin <- as.character(pois$name)
142.xpoi <- as.numeric(pois$x)
143.ypoi <- as.numeric(pois$y)
144.npoi <- length(xpoi)
145.poi.cb <- cbind(xpoi, ypoi)
146.pois
147.
148.
149.#=====
150.# Read the bathymetry
151.
152.fileName <- paste("./run_files/", project, "_bathymetry.txt", sep="")
153.fileConn <- file(fileName, "r") # open an input file connection
154.covename4 <- readLines(fileConn, n=1)
155.bathymetry <- read.table(fileConn, header=TRUE, sep=" ", comment.char = "#")
156.close(fileConn)
157.
158.
159.xbath <- bathymetry$x
160.ybath <- bathymetry$y
161.zbath <- bathymetry$z
162.bath.cb <- cbind(xbath, ybath)
163.
164.
165.#=====
166.# Read the graphics parameters
167.
168.fileName <- paste("./graphing_parameters/", project, "_graphics_parameters.txt", sep="")
169.fileConn <- file(fileName, "r") # open an input file connection
170.graph_parms <- read.table(fileConn, header=TRUE, sep="\t", comment.char = "#")
171.graph_parms
172.close(fileConn)
173.
174.deltaxleft <- as.numeric(as.character(graph_parms[1,2]))
175.deltaytop <- as.numeric(as.character(graph_parms[2,2]))
176.linespacing <- as.numeric(as.character(graph_parms[3,2]))
177.fontsize <- as.numeric(as.character(graph_parms[4,2]))
178.legendlocation <- trim(as.character(graph_parms[5,2]))
179.legendinset <- as.numeric(as.character(graph_parms[6,2]))
180.minilakex0 <- as.numeric(as.character(graph_parms[7,2]))
181.minilakey0 <- as.numeric(as.character(graph_parms[8,2]))
182.scalefactor <- as.numeric(as.character(graph_parms[9,2]))
183.
184.scalearrow <- as.numeric(as.character(graph_parms[10,2]))
185.xarrowpct <- as.numeric(as.character(graph_parms[11,2]))
186.yarrowpct <- as.numeric(as.character(graph_parms[12,2]))
187.
188.scaleline <- as.numeric(as.character(graph_parms[13,2]))
189.xscalepct <- as.numeric(as.character(graph_parms[14,2]))
190.yscalepct <- as.numeric(as.character(graph_parms[15,2]))
191.

```

```

192.
193.linespacing <- linespacing * (ymaxbox - yminbox)
194.deltaxleft <- deltaxleft * (xmaxbox - xminbox)
195.deltaytop <- deltaytop * (ymaxbox - yminbox)
196.deltaxleft
197.
198.xmin <- xminbox
199.xmax <- xmaxbox
200.ymin <- yminbox
201.ymax <- ymaxbox
202.
203.xarrow <- xmin + xarrowpct*(xmax-xmin)
204.yarrow <- ymin + yarrowpct*(ymax-ymin)
205.warrow <- scalearrow*(xmax-xmin)
206.
207.xscaleline <- xmin + xscalepct*(xmax-xmin)
208.yscaleline <- ymin + yscalepct*(ymax-ymin)
209.ndivs <- 2
210.division <- scaleline/ndivs
211.
212.
213.#=====
214.#=====
215.# Compute the spatial intervals
216.
217.x <- seq(xminbox, xmaxbox, as.integer((xmaxbox-xminbox)/nx +0.5))
218.y <- seq(yminbox, ymaxbox, as.integer((ymaxbox-yminbox)/ny +0.5))
219.
220.grdypoly <- c(xminbox, xmaxbox, xmaxbox, xminbox, xminbox)
221.grdypoly <- c(yminbox, yminbox, ymaxbox, ymaxbox, yminbox)
222.grdpoly <- cbind(grdypoly, grdypoly)
223.
224.nxy <- (nx+1)*(ny+1)
225.grdptsbox <- gridpts(grdpoly, nxy)
226.
227.
228.#=====
229.# Make a closed polygon of the cove geometry
230.
231.covelength <- length(xcove)
232.xcovpoly <- c(xcove, xcove[1])
233.ycovpoly <- c(ycove, ycove[1])
234.covpoly <- cbind(xcovpoly, ycovpoly)
235.p2 <- as(covpoly, "gpc.poly")
236.
237.
238.#=====
239.# Make a closed polygon of the bounding box geometry
240.
241.bboxlength <- length(xbox)
242.xbboxpoly <- c(xbox, xbox[1])
243.ybboxpoly <- c(ybox, ybox[1])
244.bboxpoly <- cbind(xbboxpoly, ybboxpoly)
245.p1 <- as(bboxpoly, "gpc.poly")
246.
247.
248.#=====
249.# Add the shoreline points with 0 depth
250.
251.xbath <- as.numeric(c(xbath, xcove))
252.ybath <- as.numeric(c(ybath, ycove))
253.zbath <- as.numeric(c(zbath, rep(0,length(xcove))))
254.
255.

```

```

256.#=====
257.# Set the color palette and legend data
258.
259.breakpoints <- c(0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 50, 100)
260.color_label <- c("<2 ft","<4 ft","<6 ft","<8 ft","<10 ft","<12 ft","<14 ft","<16 ft","<18 ft","<20 ft","<50 ft","<100 ft")
261.legendcolors <- c("#FF0000", "#FF7100", "#FFAA00", "#FFE300", "#4cff00", "#41de00",
262.      "#32ab00", "#00FFE3", "#00E3FF", "#00AAFF", "#0071FF", "#3900FF")
263.nlevels <- length(legendcolors)
264.
265.
266.#=====
267.#=====
268.#=====
269.# Interpolate the values with Akima without contour lines
270.
271.print("Doing Akima interpolation")
272.pdf(file=paste("./case_maps/Akima/", project, "_", resolution,"_Akima", ".pdf", sep=""), onefile=TRUE)
273.
274.akima <- interp(xbath, ybath, zbath, duplicate="strip", xo=seq(xminbox, xmaxbox, length = nx),
275.      yo=seq(yminbox, ymaxbox, length = ny), linear = TRUE, extrapol=FALSE)
276.
277.image(akima, xlim = extendrange(xbox), ylim = extendrange(ybox), breaks=breakpoints,
278.      col=legendcolors, lwd=0.05, lty=3, xlab="Easting (ft)", ylab="Northing(ft)")
279.
280.#contour(akima, add=TRUE, labcex=0.7, drawlabels=TRUE)      # contours
281.points(xdock, ydock, cex=0.7, pch=15, col="white")          # Boat slips - square outline
282.points(xdock, ydock, cex=0.3, pch=20, col="black")         # Boat slips - black dot
283.plot(setdiff(p1, p2), poly.args = list(col = "grey"), add = TRUE)
284.lines(xcovepoly, ycovepoly, type="l", lwd=2, col="blueviolet") # Cove outline
285.if(npoi != 0){
286.      points(xpoi, ypoi, cex=1.0, pch=17, col="black")      # Points of Interest - black triangle
287.      text(xpoi, ypoi, poin, pos=4, offset=0.25, cex=0.65)  # Label the points of interest
288.}
289.
290.# To add various elements is done on the basis of %x and %y of the location of the lower left-hand corner of the element
291.# The overall size is uniformly adjusted according to a scale factor that applies to each element
292.
293.usr <- par( "usr" )
294.text(usr[1]+deltaxleft, usr[4]-deltaytop, paste("PLV: Image created with R:",Sys.Date()), pos=4, cex=fontsize)
295.text(usr[1]+deltaxleft, usr[4]-deltaytop-1*linespacing, paste("Interpolations with Akima"), pos=4, cex=fontsize)
296.text(usr[1]+deltaxleft, usr[4]-deltaytop-2*linespacing, paste("Area of the cove:",areap,"acres."), pos=4, cex=fontsize)
297.text(usr[1]+deltaxleft, usr[4]-deltaytop-3*linespacing, paste("Shore length of the cove:",miles,"miles."), pos=4, cex=fontsize)
298.text(usr[1]+deltaxleft, usr[4]-deltaytop-4*linespacing, paste("Spatial resolution:",resolution,"ft."), pos=4, cex=fontsize)
299.text(usr[1]+deltaxleft, usr[4]-deltaytop-5*linespacing, paste(nx,"by",ny,"grid."), pos=4, cex=fontsize)
300.text(usr[1]+deltaxleft, usr[4]-deltaytop-6*linespacing, paste("White squares & black dots - boat slips"), pos=4, cex=fontsize)
301.text(usr[1]+deltaxleft, usr[4]-deltaytop-7*linespacing, paste("Depths referenced to 2,462 ft ASL"), pos=4, cex=fontsize)
302.
303.
304.north.arrow(xarrow,yarrow,warrow,lab="NORTH",cex.lab=0.5,tcol="black",col="orange")
305.map.scale(xscaleline,yscaleline,scaleline,"ft",ndivs,subdiv=division,tcol="black",scol="black",sfc="black")
306.
307.title(paste("A Bathymetric Map of ", areaname), font = 4)
308.
309.legend(legendlocation, color_label, col=legendcolors, lwd=10, title="Contours", cex=fontsize, inset=legendinset)
310.
311.
312.#=====
313.# Plot the lake and the cove in small version
314.
315.shore <- read.table(paste("./basic_inputs/DCL_shoreline_SPSC_V2.txt", sep="\t"), header=TRUE, sep="\t", comment.char = "#")
316.xlake <- shore$Easting
317.ylake <- shore$Northing
318.
319.# Check to see if closed polygon. If not make it that way.

```

```

320.if((xlake[1] != xlake[length(xlake)]) || (ylake[1] != ylake[length(ylake)])){
321.      xlake[length(xlake)+1] = xlake[1]
322.      ylake[length(ylake)+1] = ylake[1]
323.}
324.
325.# Compute the plot scaling factors
326.xmin <- min(xlake)
327.xmax <- max(xlake)
328.ymin <- min(ylake)
329.ymax <- max(ylake)
330.
331.scalex <- (xmaxbox-xminbox)/(xmax-xmin)
332.scaley <- (ymaxbox-yminbox)/(ymax-ymin)
333.
334.
335.# Read the file with all of the cove locations
336.fileConn <- file("./basic_inputs/ListOfCoves.txt", "rt")
337.covelocations <- read.table(fileConn, header=TRUE, sep="\t", comment.char = "#")
338.close(fileConn)
339.
340.M <- length(covelocations$ID)
341.
342.# Search for the area name to be projected on the lake map - stored as "k"
343.narea <- 0
344.for(j in 1:M){
345.      if(as.character(covelocations$ID[j]) == project) narea <- j
346.}
347.if(narea != 0){
348.      x00 <- xminbox + minilakex0*(xmaxbox - xminbox)          # x0-coordinate of small box
349.      y00 <- yminbox + minilakey0*(ymaxbox - yminbox)          # y0-coordinate of small box
350.      smallcovex <- x00 + scalefactor*scalex*(covelocations$Easting[narea] - xmin)
351.      smallcovey <- y00 + scalefactor*scaley*(covelocations$Northing[narea] - ymin)
352.
353.      xbb <- c(xmin, xmax, xmax, xmin, xmin)
354.      ybb <- c(ymin, ymin, ymax, ymax, ymin)
355.      lines(x00 + scalefactor*scalex*(xlake-xmin), y00 + scalefactor*scaley*(ylake-ymin), type="l", col="red",
lwd=1)
          # Lake outline
356.      lines(x00 + scalefactor*scalex*(xbb-xmin), y00 + scalefactor*scaley*(ybb-ymin), type="l", col="blue",
lwd=1)
          # Bounding box
357.
358.
359.      polygon(x00+scalefactor*scalex*(xbb-xmin), y00+scalefactor*scaley*(ybb-ymin), col="gold")
360.      polygon(x00+scalefactor*scalex*(xlake-xmin), y00+scalefactor*scaley*(ylake-ymin), col="light blue")
361.      points(smallcovex, smallcovey, cex=1.2, col="red")          # Circle of cove on the minnie whole lake map
362.
363.}
364.
365.dev.off()
366.
367.#=====
368.#=====
369.# Interpolate the values with Akima and contours
370.
371.print("Doing Akima interpolation")
372.pdf(file=paste("./case_maps/Akima2/", project, "_", resolution, "_Akima", ".pdf", sep=""), onefile=TRUE)
373.
374.image(akima, xlim = extendrange(xbox), ylim = extendrange(ybox), breaks=breakpoints,
375.      col=legendcolors, lwd=0.05, lty=3, xlab="Easting (ft)", ylab="Northing(ft)")
376.
377.contour(akima, add=TRUE, labcex=0.7, drawlabels=TRUE)          # contours
378.lines(xcovpoly, ycovpoly, type="l", lwd=3, col=96)          # Cove outline
379.points(xdock, ydock, cex=0.7, pch=15, col="white")          # Boat slips - square outline
380.points(xdock, ydock, cex=0.3, pch=20, col="black")          # Boat slips - black dot
381.plot(setdiff(p1, p2), poly.args = list(col = "grey"), add = TRUE)

```

```

382.if(npoi != 0){
383.      points(xpoi, ypoi, cex=1.0, pch=17, col="black")      # Points of Interest - black triangle
384.      text(xpoi, ypoi, poin, pos=4, offset=0.25, cex=0.65)  # Label the points of interest
385.}
386.
387.usr <- par( "usr" )
388.text(usr[1]+deltaxleft, usr[4]-deltaytop, paste("PLV: Image created with R:", Sys.Date()), pos=4, cex=fontsize)
389.text(usr[1]+deltaxleft, usr[4]-deltaytop-1*linespacing, paste("Interpolations with Akima"), pos=4, cex=fontsize)
390.text(usr[1]+deltaxleft, usr[4]-deltaytop-2*linespacing, paste("Area of the cove:", areap, "acres."), pos=4, cex=fontsize)
391.text(usr[1]+deltaxleft, usr[4]-deltaytop-3*linespacing, paste("Shore length of the cove:", miles, "miles."), pos=4, cex=fontsize)
392.text(usr[1]+deltaxleft, usr[4]-deltaytop-4*linespacing, paste("Spatial resolution:", resolution, "ft."), pos=4, cex=fontsize)
393.text(usr[1]+deltaxleft, usr[4]-deltaytop-5*linespacing, paste(nx, "by", ny, "grid."), pos=4, cex=fontsize)
394.text(usr[1]+deltaxleft, usr[4]-deltaytop-6*linespacing, paste("White squares & black dots - boat slips"), pos=4, cex=fontsize)
395.text(usr[1]+deltaxleft, usr[4]-deltaytop-7*linespacing, paste("Depths referenced to 2,462 ft ASL"), pos=4, cex=fontsize)
396.
397.north.arrow(xarrow,yarrow,warrow,lab="NORTH",cex.lab=0.5,tcol="black",col="orange")
398.map.scale(xscaleline,yscaleline,scaleline,"ft",ndivs,subdiv=division,tcol="black",scol="black",sfc="black")
399.
400.title(paste("A Bathymetric Map of ", areaname), font = 4)
401.
402.legend(legendlocation, color_label, col=legendcolors, lwd=10, title="Contours", cex=fontsize, inset=legendinset)
403.
404.
405.#=====
406.# Plot the lake and the cove in small version
407.
408.if(narea != 0){
409.      x00 <- xminbox + minilakex0*(xmaxbox - xminbox)      # x0-coordinate of small box
410.      y00 <- yminbox + minilakey0*(ymaxbox - yminbox)      # y0-coordinate of small box
411.      smallcovex <- x00 + scalefactor*scalex*(covelocations$Easting[narea] - xmin)
412.      smallcovex <- y00 + scalefactor*scaley*(covelocations$Northing[narea] - ymin)
413.
414.      xbb <- c(xmin, xmax, xmax, xmin, xmin)
415.      ybb <- c(ymin, ymin, ymax, ymax, ymin)
416.      lines(x00 + scalefactor*scalex*(xlake-xmin), y00 + scalefactor*scaley*(ylake-ymin), type="l", col="red",
lwd=1)      # Lake outline
417.      lines(x00 + scalefactor*scalex*(xbb-xmin), y00 + scalefactor*scaley*(ybb-ymin), type="l", col="blue",
lwd=1)      # Bounding box
418.
419.
420.      polygon(x00+scalefactor*scalex*(xbb-xmin), y00+scalefactor*scaley*(ybb-ymin), col="gold")
421.      polygon(x00+scalefactor*scalex*(xlake-xmin), y00+scalefactor*scaley*(ylake-ymin), col="light blue")
422.      points(smallcovex, smallcovex, cex=1.2, col="black")      # Circle of cove on the minnie whole lake map
423.
424.}
425.
426.dev.off()
427.
428.#=====
429.#=====
430.# Interpolate the values with Loess
431.
432.print("Doing Loess interpolation")
433.pdf(file=paste("../case_maps/Loess/", project, "_", resolution, "_Loess", ".pdf", sep=""), onefile=TRUE)
434.
435.parm <- 0.02      # parm = 'span' variable, a value between 0 and 1. Typically span=0.05
436.
437.# Resulting plot to be written to a pdf file.
438.
439.loess <- interp.loess(xbath, ybath, zbath, gridlen = c(nx,ny), span=parm)
440.
441.image(loess, xlim = extendrange(xbox), ylim = extendrange(ybox), breaks=breakpoints,
442.      col=legendcolors, lwd=0.05, lty=3, xlab="Easting (ft)", ylab="Northing(ft)")
443.

```



```

444.lines(xcovpoly, ycovpoly, type="l", lwd=3, col=96)           # Cove outline
445.points(xdock, ydock, cex=0.7, pch=15, col="white")        # Boat slips - square outline
446.points(xdock, ydock, cex=0.3, pch=20, col="black")        # Boat slips - black dot
447.plot(setdiff(p1, p2), poly.args = list(col = "grey"), add = TRUE)
448.if(npoi != 0){
449.    points(xpoi, ypoi, cex=1.0, pch=17, col="black")        # Points of Interest - black triangle
450.    text(xpoi, ypoi, poin, pos=4, offset=0.25, cex=0.65)    # Label the points of interest
451.}
452.
453.usr <- par( "usr" )
454.text(usr[1]+deltaxleft, usr[4]-deltaytop, paste("PLV: Image created with R:", Sys.Date()), pos=4, cex=fontsize)
455.text(usr[1]+deltaxleft, usr[4]-deltaytop - linespacing, paste("Interpolations with Loess"), pos=4, cex=fontsize)
456.text(usr[1]+deltaxleft, usr[4]-deltaytop-2*linespacing, paste("Area of the cove:", areap, "acres."), pos=4, cex=fontsize)
457.text(usr[1]+deltaxleft, usr[4]-deltaytop-3*linespacing, paste("Shore length of the cove:", miles, "miles."), pos=4, cex=fontsize)
458.text(usr[1]+deltaxleft, usr[4]-deltaytop-4*linespacing, paste("Spatial resolution:", resolution, "ft."), pos=4, cex=fontsize)
459.text(usr[1]+deltaxleft, usr[4]-deltaytop-5*linespacing, paste(nx, "by", ny, "grid."), pos=4, cex=fontsize)
460.text(usr[1]+deltaxleft, usr[4]-deltaytop-6*linespacing, paste("White squares & black dots - boat slips"), pos=4, cex=fontsize)
461.text(usr[1]+deltaxleft, usr[4]-deltaytop-7*linespacing, paste("Depths referenced to 2,462 ft ASL"), pos=4, cex=fontsize)
462.
463.north.arrow(xarrow, yarrow, warrow, lab="NORTH", cex.lab=0.5, tcol="black", col="orange")
464.map.scale(xscaleline, yscaleline, scaleline, "ft", ndivs, subdiv=division, tcol="black", scol="black", sfc="black")
465.
466.title(paste("A Bathymetric Map of ", areaname), font = 4)
467.
468.legend(legendlocation, color_label, col=legendcolors, lwd=10, title="Contours", cex=fontsize, inset=legendinset)
469.
470.
471.#=====
472.# Plot the lake and the cove in small version
473.
474.if(narea != 0){
475.    x00 <- xminbox + minilake0*(xmaxbox - xminbox)           # x0-coordinate of small box
476.    y00 <- yminbox + minilake0*(ymaxbox - yminbox)           # y0-coordinate of small box
477.    smallcovex <- x00 + scalefactor*scalex*(covelocations$Easting[narea] - xmin)
478.    smallcovex <- y00 + scalefactor*scaley*(covelocations$Northing[narea] - ymin)
479.
480.    xbb <- c(xmin, xmax, xmax, xmin, xmin)
481.    ybb <- c(ymin, ymin, ymax, ymax, ymin)
482.    lines(x00 + scalefactor*scalex*(xlake-xmin), y00 + scalefactor*scaley*(ylake-ymin), type="l", col="red",
lwd=1)
483.    lines(x00 + scalefactor*scalex*(xbb-xmin), y00 + scalefactor*scaley*(ybb-ymin), type="l", col="blue",
lwd=1)
484.
485.    polygon(x00+scalefactor*scalex*(xbb-xmin), y00+scalefactor*scaley*(ybb-ymin), col="gold")
486.    polygon(x00+scalefactor*scalex*(xlake-xmin), y00+scalefactor*scaley*(ylake-ymin), col="light blue")
487.
488.    points(smallcovex, smallcovex, cex=1.2, col="black")      # Circle of cove on the minnie whole lake map
489.
490.}
491.
492.
493.dev.off()
494.#=====
495.begTime
496.runTime <- Sys.time()-begTime
497.runTime
498.print(sprintf("Finished with -%s- in -CoveMaps.R-", project))
499.warnings()

```