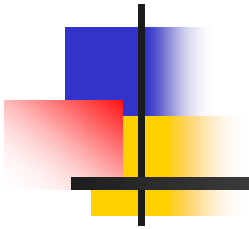
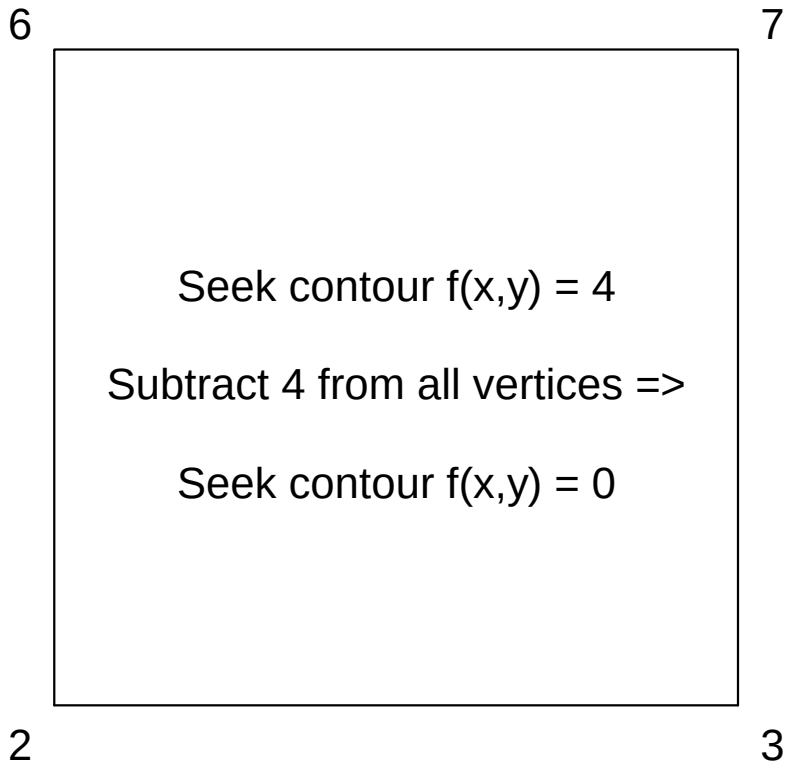
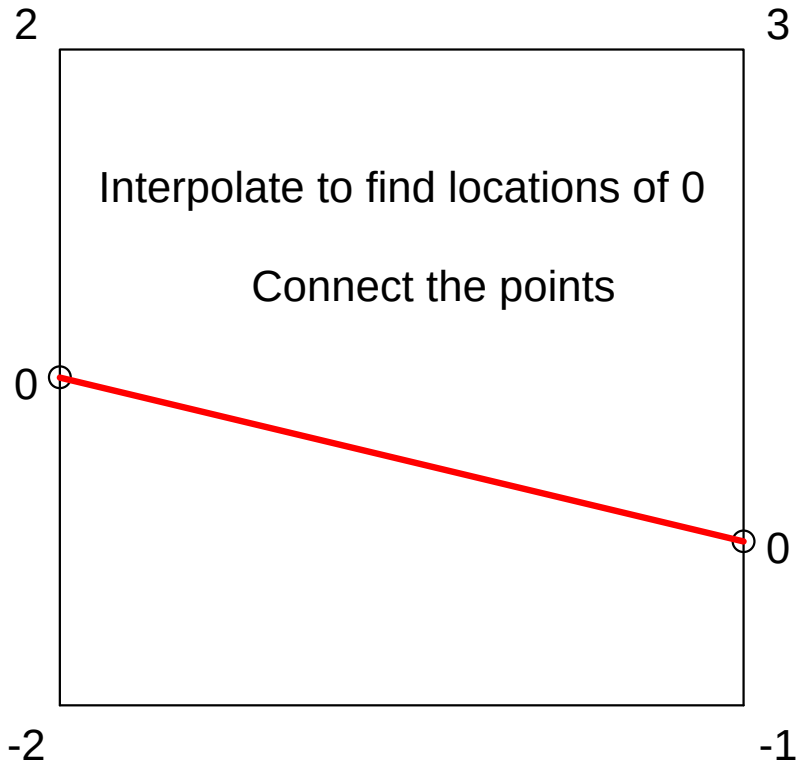
Rectangular Grid Contouring f(x,y) = c



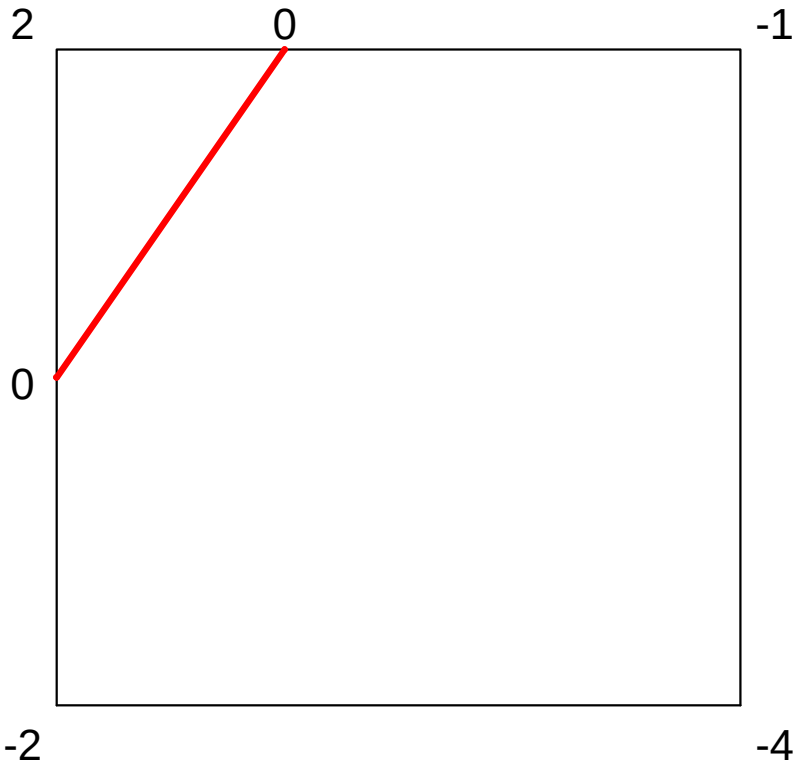
- Subtract c from all grid values => g(x,y)
- Find contour for g(x,y) = 0
- For each rectangle
  - Skip if all vertices have the same sign
  - For edges whose vertices have opposite sign (-,+)
    - Linearly interpolate to find location of 0 on such edges
    - If two locations, draw connecting segment and move on
    - If four locations, draw two connecting segments joining left most pair and right most pair

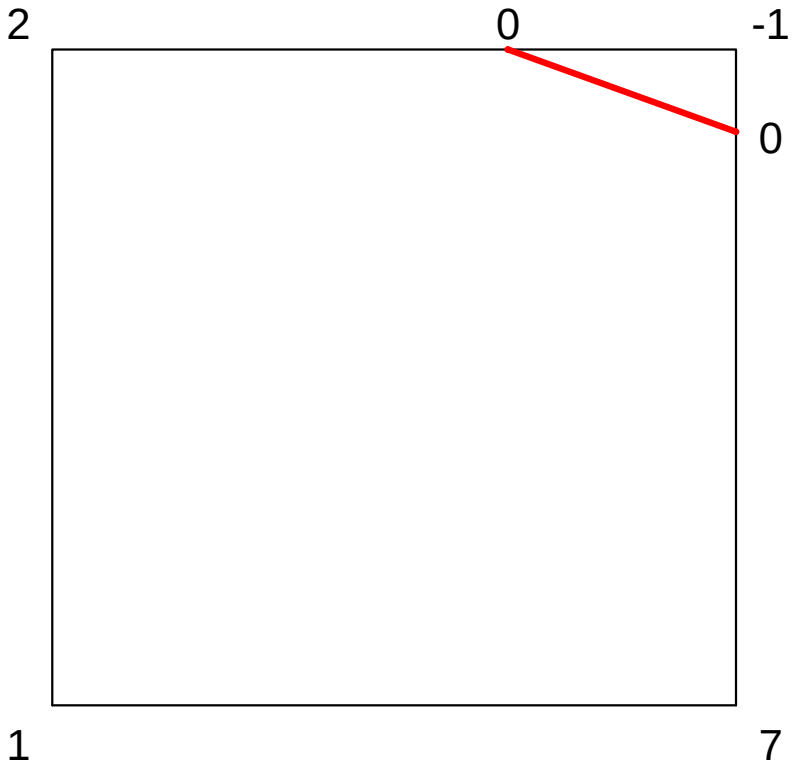
Case 1: Opposite Edges



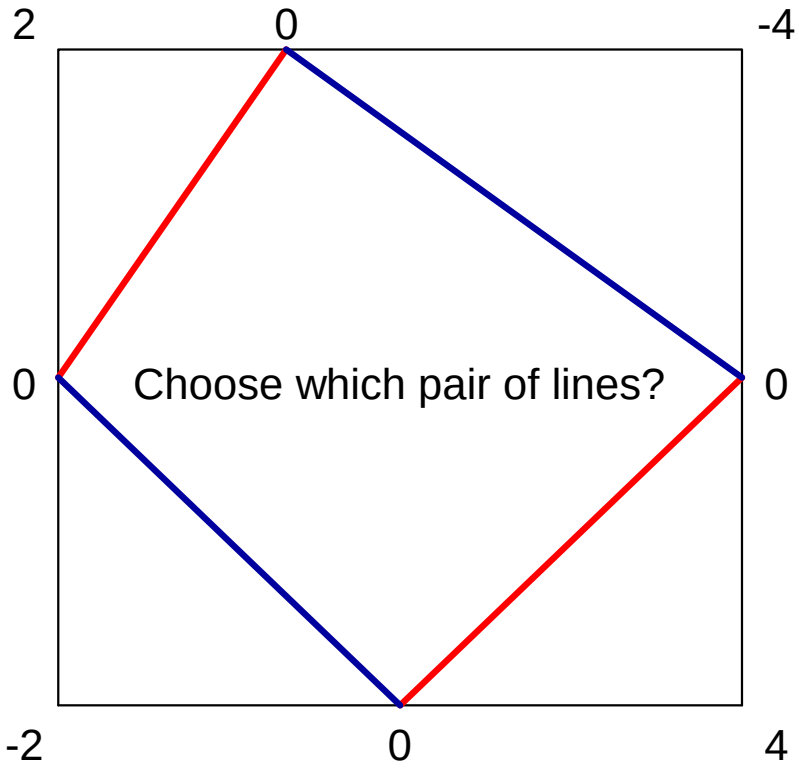


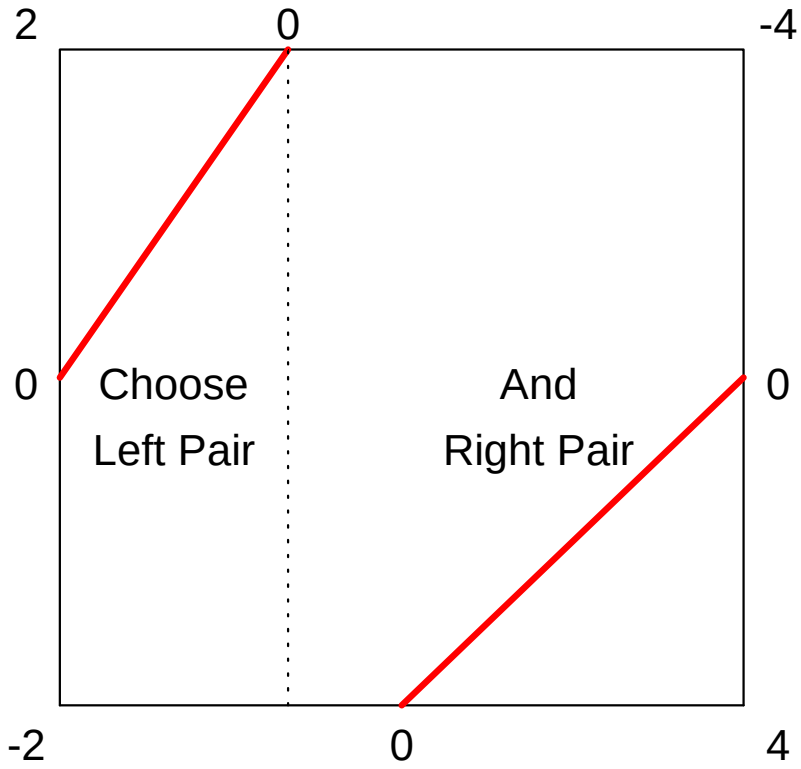
Case 2: Adjacent Edges





Case 3:
All Four Edges





Four Edge Solution
Grosse's Method

- Fit bilinear model to the four vertices
  - v= a1 + a2*x + a3*y + a4*x*y
  - Use model contours +> connect small x pairs and large x pairs

x	y	v
0	0	-2
0	1	2
1	1	-4
1	0	4

-2	1	0	0	0	a1
2	1	0	1	0	a2
-4	1	1	1	1	a3
4	1	1	0	0	a4

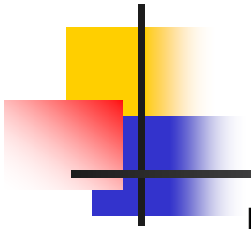
a1=-2

a2=6

a3=4

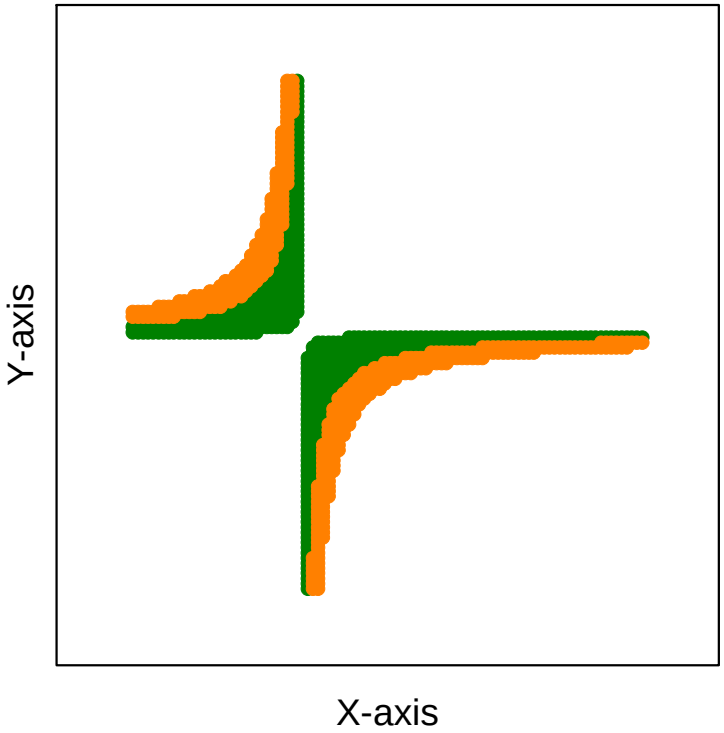
a4=-12

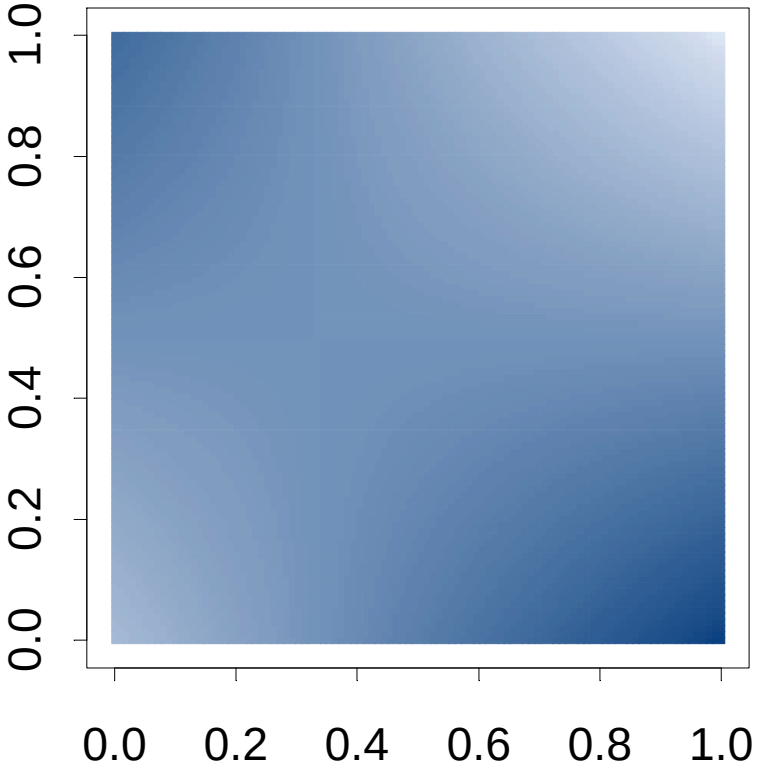
Contours of a Bilinear Model
A Quick Look



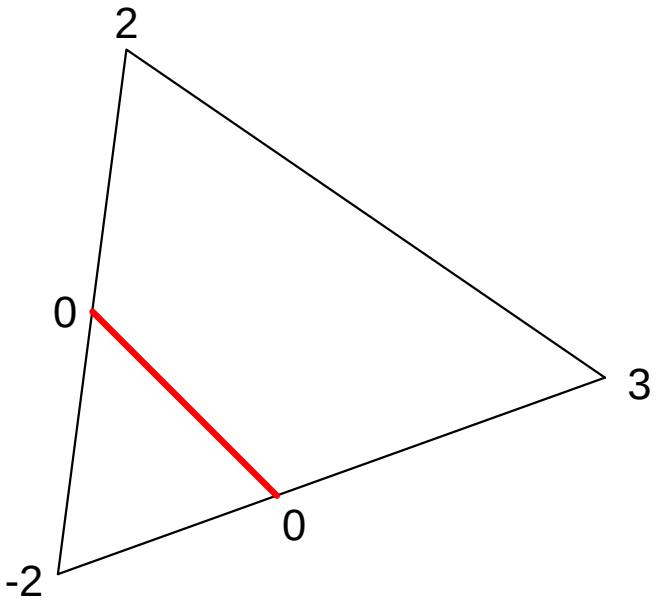
```
n = 100
x = seq(0,1,length=100)
mat = as.matrix(expand.grid(list(x=x,y=x)))

xm = mat[,1]; ym = mat[,2]
a1 = -2; a2 = 6; a3 = 4; a4 = -12
vals = a1 + a2*xm + a3*ym + a4*xm*ym

plot(c(0,1),c(0,1),type='n')
   good = abs(vals-.05) <= .045   #  near .05
       points(mat[good,],col=4,cex=.6,pch=16) # green
   good = abs(vals-.15) <= .045   # near .15
      points(mat[good,],col=5,cex=.6,pch=16)  # orange

graphsheet()
image(x,x,matrix(vals,ncol=n))
```

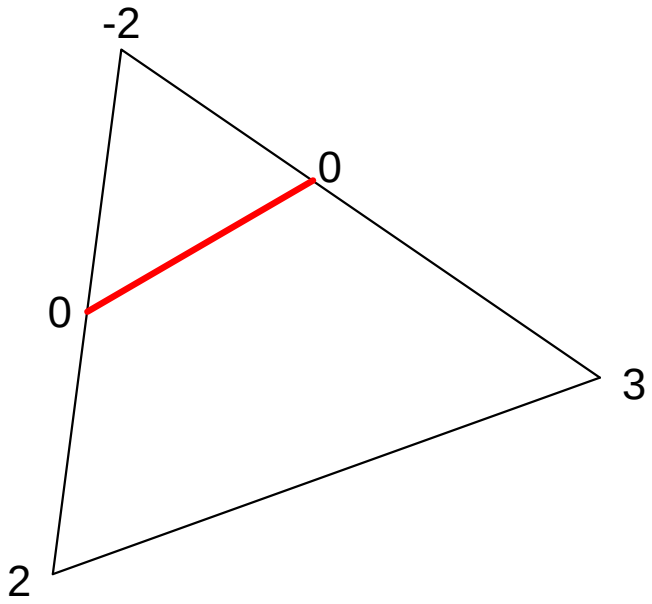
Regions larger than zero: Top left and lower right





Contour Segments for Triangles





Contour Tracking By Flipping Triangles

- Can also flip the vertex opposite the crossed face for a simplex in higher dimensions: See Alan Wilks paper