

Math 534 Homework 1 (Part 1) - 25 points

Problem 1: Graphing a bivariate function and its second order approximation

Use R to graph a 3-d plot of the function $f(x_1, x_2) = \cos(x_1 x_2)$ and its second order Taylor approximation $h(x_1, x_2) = 1 - \frac{\pi^2}{8} x_1^2$. Following are the requirements:

- i. [1 pt] Write an R function that computes the function f for any input x_1 , and x_2 .

```
f <- function(x1,x2){cos(x1*x2)}
```

- ii. [1 pt] Write an R function that computes the function h for any input x_1 , and x_2 .

```
h <- function(x1,x2){1-pi^2/8*x1^2} #including x2 because it is in the problem
```

- iii. [3 pts] Use the R functions in (i) and (ii) to plot a three dimensional graph that contains the surfaces for both f and h on the same frame (axes). Color surface of f red and surface h blue. The values of x_1 for your graph should range in the interval $[-\frac{\pi}{4}, \frac{\pi}{4}]$ and the values of x_2 must range in the interval $[\frac{\pi}{4}, \frac{3\pi}{4}]$. Divide each of the intervals by 30 equal points and evaluate the function on the 30 by 30 grid generated, and then make your plot.

```
library(rgl)

x1 <- seq(from=-pi/4,to=pi/4,by=(pi/4-(-pi/4))/29)
x2 <- seq(from=pi/4,to=3*pi/4,by=(3*pi/4-(pi/4))/29)

f <- outer(x1, x2, FUN = f)
persp3d(x1, x2, f, col = "red", alpha = 0.5)

h <- outer(x1, x2, FUN = h)
persp3d(x1, x2, h, col = "blue", alpha = 0.5, add = TRUE)

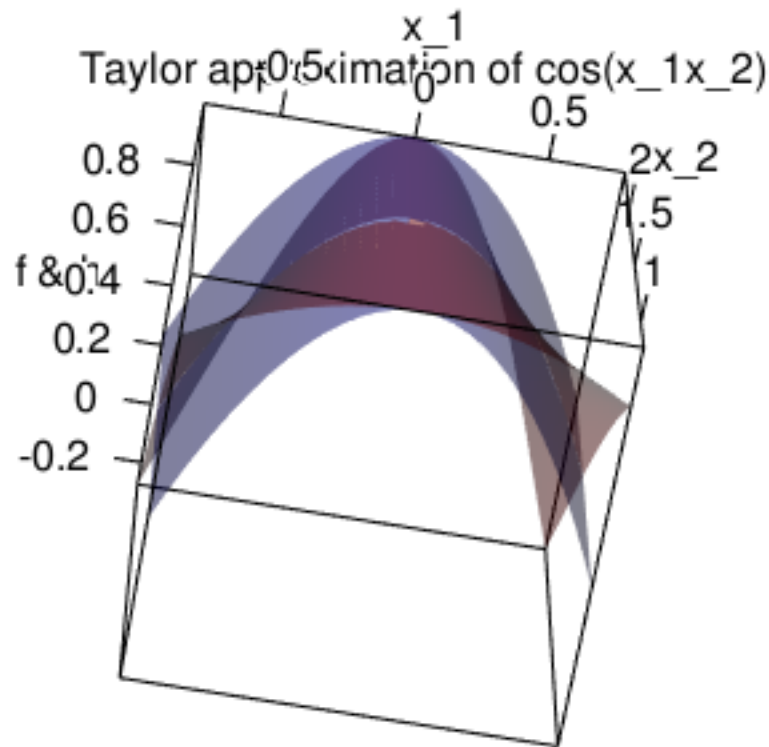
#rglwidget(controllers = ) # This is to get it to print in pdf
```

- iv. [1 pt] Label the first axis x_1 and the second axis x_2 . Your graph should have the title “Taylor approximation of $\cos(x_1 x_2)$.” Make sure to show the plot from an angle with a good view of the function.

```
persp3d(x1, x2, f, col = "red", alpha = 0.5,
        xlab = "x_1", ylab = "x_2", zlab = "f & h",
        main = "Taylor approximation of cos(x_1x_2)")
persp3d(x1, x2, h, col = "blue", alpha = 0.5, add = TRUE)
view3d(theta = -10, phi = -65, fov = 10, zoom = .9)

rglwidget(controllers = ) # This is to get it to print in pdf
```

```
## cleared error 1285
```



- v. [3 pts] In a separate 3d plot, graph the absolute value error function $e(x_1, x_2) = |f(x_1, x_2) - h(x_1, x_2)|$. Use the same range and grid for x_1 and x_2 as described in (iii).

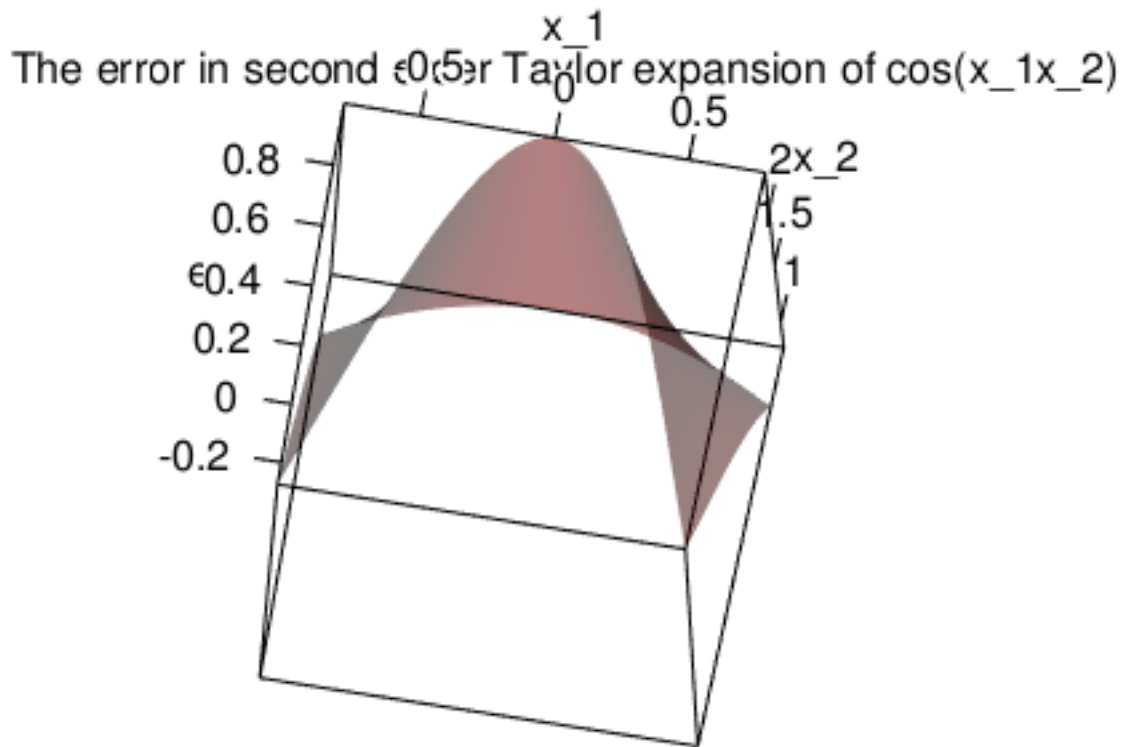
```
e <- abs(f-h)
persp3d(x1, x2, f, col = "red", alpha = 0.5,
        xlab = "x_1", ylab = "x_2", zlab = "e")
```

- vi. [1 pt] Label the first axis x_1 and the second axis x_2 . Title your plot “The error in second order Taylor expansion of $\cos(x_1x_2)$.”

```
e <- abs(f-h)
persp3d(x1, x2, f, col = "red", alpha = 0.5,
        xlab = "x_1", ylab = "x_2", zlab = "e",
        main = "The error in second order Taylor expansion of cos(x_1x_2)")
```

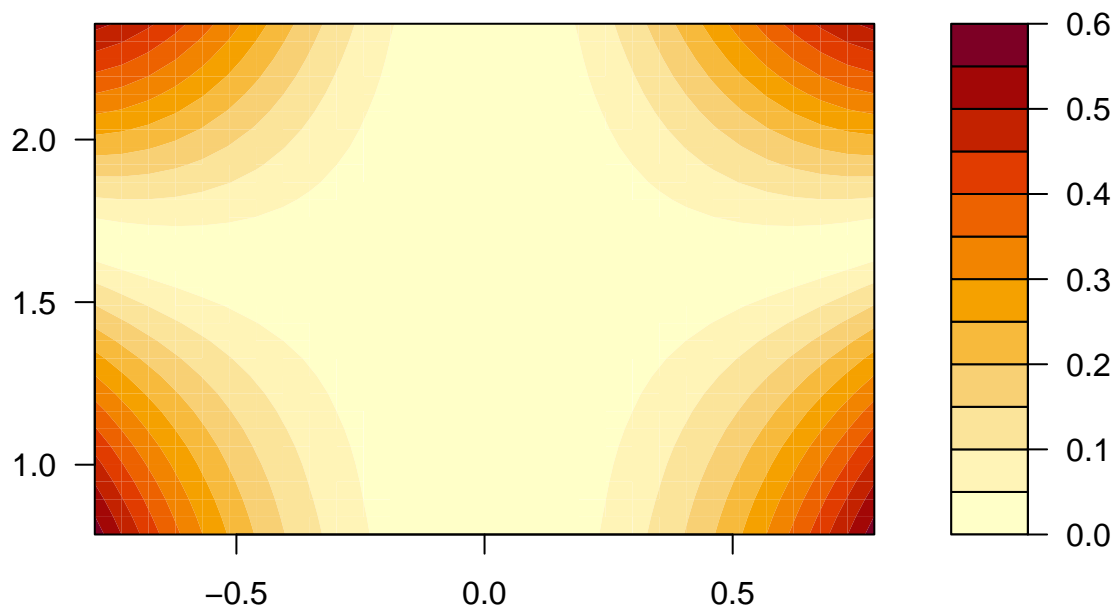
```
view3d(theta = -10, phi = -65, fov = 10, zoom = .9)
rglwidget(controllers = ) # This is to get it to print in pdf
```

```
## cleared error 1285
```



- vii. [3 pts] Plot the constant value contours for $e(x_1, x_2)$, and explain the magnitude of the error at various locations in the context of the problem.

```
filled.contour(x1, x2, e)
```



```
#grid(nx = 20, ny = 20)
```

Problem 2: Given a $p \times 1$ vector μ and a $p \times p$ positive definite matrix Σ , the pdf for a p -variate normal

density at a point $\mathbf{x} = (x_1, x_2, \dots, x_p)^T$ is given by

$$f(\mathbf{x}) = (2\pi)^{-\frac{p}{2}} |\Sigma|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\right].$$

Now consider the bivariate normal random variable, where

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix}.$$

(a) [3 pts] Write the second order Taylor expansion for $f(\mathbf{x})$, for the bivariate normal density, **around** the point

$$\mathbf{x}_0 = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}.$$

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + (\mathbf{x} - \mathbf{x}_0)^T \nabla f(\mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \nabla^2 f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$$

$$f(\mathbf{x}) = \frac{1}{2\pi\sqrt{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}} \exp\left[-\frac{1}{2} \frac{(\sigma_{22}(x_1 - \mu_1)^2 - (\sigma_{21} + \sigma_{12})(x_1 - \mu_1)(x_2 - \mu_2) + \sigma_{11}(x_2 - \mu_2)^2)}{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}\right]$$

$$\frac{\partial}{\partial x_1} f(\mathbf{x}) = \frac{1}{2\pi\sqrt{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}} \exp\left[-\frac{1}{2} \frac{(\sigma_{22}(x_1 - \mu_1)^2 - (\sigma_{21} + \sigma_{12})(x_1 - \mu_1)(x_2 - \mu_2) + \sigma_{11}(x_2 - \mu_2)^2)}{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}\right] \cdot \left[\frac{-1}{2(\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21})}\right]$$

$$\frac{\partial}{\partial x_2} f(\mathbf{x}) = \frac{1}{2\pi\sqrt{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}} \exp\left[-\frac{1}{2} \frac{(\sigma_{22}(x_1 - \mu_1)^2 - (\sigma_{21} + \sigma_{12})(x_1 - \mu_1)(x_2 - \mu_2) + \sigma_{11}(x_2 - \mu_2)^2)}{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}\right] \cdot \left[\frac{-1}{2(\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21})}\right]$$

$$\frac{\partial^2}{\partial x_1 \partial x_1} f(\mathbf{x}) = \frac{\exp\left[-\frac{1}{2} \frac{(\sigma_{22}(x_1 - \mu_1)^2 - (\sigma_{21} + \sigma_{12})(x_1 - \mu_1)(x_2 - \mu_2) + \sigma_{11}(x_2 - \mu_2)^2)}{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}\right]}{2\pi\sqrt{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}} \left[\left[\frac{-1}{2(\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21})}\right](2\sigma_{22}(x_1 - \mu_1) - (\sigma_{12} + \sigma_{21}))(x_2 - \mu_2)\right]$$

$$\frac{\partial^2}{\partial x_2 \partial x_2} f(\mathbf{x}) = \frac{\exp\left[-\frac{1}{2} \frac{(\sigma_{22}(x_1 - \mu_1)^2 - (\sigma_{21} + \sigma_{12})(x_1 - \mu_1)(x_2 - \mu_2) + \sigma_{11}(x_2 - \mu_2)^2)}{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}\right]}{2\pi\sqrt{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}} \left[\left[\frac{-1}{2(\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21})}\right](2\sigma_{11}(x_2 - \mu_2) - (\sigma_{12} + \sigma_{21}))(x_1 - \mu_1)\right]$$

$$\frac{\partial^2}{\partial x_1 \partial x_2} f(\mathbf{x}) = \frac{\exp\left[-\frac{1}{2} \frac{(\sigma_{22}(x_1 - \mu_1)^2 - (\sigma_{21} + \sigma_{12})(x_1 - \mu_1)(x_2 - \mu_2) + \sigma_{11}(x_2 - \mu_2)^2)}{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}\right]}{2\pi\sqrt{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}} \left[\left[\frac{(2\sigma_{11}(x_2 - \mu_2) - (\sigma_{12} + \sigma_{21}))(x_1 - \mu_1)}{4(\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21})^2}\right]\right]$$

$$f(\mathbf{x}_0) = \frac{1}{2\pi\sqrt{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}}$$

$$\frac{\partial}{\partial x_1} f(\mathbf{x}_0) = \frac{1}{2\pi\sqrt{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}}$$

$$\frac{\partial}{\partial x_2} f(\mathbf{x}_0) = \frac{1}{2\pi\sqrt{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}}$$

$$\frac{\partial^2}{\partial x_1 \partial x_1} f(\mathbf{x}_0) = \frac{-\sigma_{22}}{2\pi(\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21})^{3/2}}$$

$$\frac{\partial^2}{\partial x_2 \partial x_2} f(\mathbf{x}_0) = \frac{-\sigma_{11}}{2\pi(\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21})^{3/2}}$$

$$\frac{\partial^2}{\partial x_1 \partial x_2} f(\mathbf{x}_0) = \frac{(\sigma_{12} + \sigma_{21})}{2 \cdot 2\pi(\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21})^{3/2}}$$

$$f(\mathbf{x}) \approx \frac{1}{2\pi\sqrt{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}} + \frac{x_1 - \mu_1 + x_2 - \mu_2}{2\pi\sqrt{\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21}}} + \frac{\sigma_{11}(x_1 - \mu_1)^2 - (\sigma_{12} + \sigma_{21})(x_1 - \mu_1)(x_2 - \mu_2) + \sigma_{22}(x_2 - \mu_2)^2}{4\pi(\sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21})^{3/2} \left(\sigma_{11}\sigma_{22} - \frac{(\sigma_{12} + \sigma_{21})^2}{4} \right)}$$

(b) [3 pts] Graph a 3-d plot of the function $f(x_1, x_2)$ and its second order Taylor expansion for the following parameters (for each set of parameters, f and its approximation should be on the same frame):

$$(i) \mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma = \begin{pmatrix} 1 & -0.3 \\ -0.3 & 1 \end{pmatrix}; (ii) \mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$$

Note that the means and variance for each of the variables in the cases (i) and (ii) are 0 and 1 respectively, To get a reasonable graph, use the interval $[-1.5, 1.5]$ for the (x_1, x_2) grid.

```
#bivariate normal distribution density function
f <- function(x1,x2,mu1,mu2,sigma11,sigma12,sigma21,sigma22){
  1/(2*pi*sqrt(sigma11*sigma22-sigma12*sigma21))*exp(-((sigma22*(x1-mu1)^2)-(sigma12+sigma21)*(x1-mu1)*
})

#second order Taylor expansion of f
h <- function(x1,x2,mu1,mu2,sigma11,sigma12,sigma21,sigma22){
  1/(2*pi*sqrt(sigma11*sigma22-sigma12*sigma21)) + (x1-mu1+x2-mu2)/(2*pi*sqrt(sigma11*sigma22-sigma12*sigma21))
}
```

(i)

```
x1 <- seq(from=-1.5,to=1.5,by=(1.5-(-1.5))/29)
x2 <- seq(from=-1.5,to=1.5,by=(1.5-(-1.5))/29)

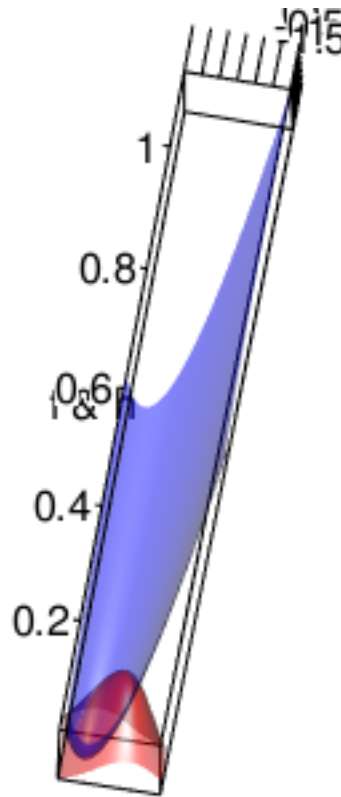
f_i <- outer(x1, x2,mu1 = 0, mu2 = 0, sigma11 = 1, sigma12 = -0.3, sigma21 = -0.3,sigma22 = 1, FUN = f)
h_i <- outer(x1, x2,mu1 = 0, mu2 = 0, sigma11 = 1, sigma12 = -0.3, sigma21 = -0.3,sigma22 = 1, FUN = h)

persp3d(x1, x2, f_i, col = "red", alpha = 0.5,
        xlab = "x_1", ylab = "x_2", zlab = "f & h",
        main = "Taylor approximation")
persp3d(x1, x2, h_i, col = "blue", alpha = 0.5, add = TRUE)
view3d(theta = -10, phi = -65, fov = 10, zoom = .9)

rglwidget(controllers = ) # This is to get it to print in pdf

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package; using rgl.snapshot() instead

## cleared error 1285
```



(ii)

```
#x1 <- seq(from=-1.5,to=1.5,by=(1.5-(-1.5))/29)
#x2 <- seq(from=-1.5,to=1.5,by=(1.5-(-1.5))/29)

f_ii <- outer(x1, x2,mu1 = 0, mu2 = 0, sigma11 = 1, sigma12 = 0.8, sigma21 = 0.8,sigma22 = 1, FUN = f)
h_ii <- outer(x1, x2,mu1 = 0, mu2 = 0, sigma11 = 1, sigma12 = 0.8, sigma21 = 0.8,sigma22 = 1, FUN = h)

persp3d(x1, x2, f_ii, col = "red", alpha = 0.5,
        xlab = "x_1", ylab = "x_2", zlab = "f & h",
        main = "Taylor approximation")
persp3d(x1, x2, h_ii, col = "blue", alpha = 0.5, add = TRUE)
view3d(theta = -10, phi = -65, fov = 10, zoom = .9)

rglwidget(controllers = ) # This is to get it to print in pdf

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package; using rgl.snapshot() instead

## cleared error 1285
```



(iii) Extra

```
#x1 <- seq(from=-1.5,to=1.5,by=(1.5-(-1.5))/29)
#x2 <- seq(from=-1.5,to=1.5,by=(1.5-(-1.5))/29)

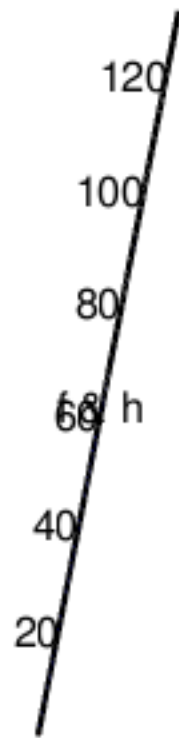
f_iii <- outer(x1, x2,mu1 = 0, mu2 = 0, sigma11 = 1, sigma12 = 0.95, sigma21 = 0.95,sigma22 = 1, FUN = :
h_iii <- outer(x1, x2,mu1 = 0, mu2 = 0, sigma11 = 1, sigma12 = 0.95, sigma21 = 0.95,sigma22 = 1, FUN = :

persp3d(x1, x2, f_iii, col = "red", alpha = 0.5,
        xlab = "x_1", ylab = "x_2", zlab = "f & h",
        main = "Taylor approximation")
persp3d(x1, x2, h_iii, col = "blue", alpha = 0.5, add = TRUE)
view3d(theta = -10, phi = -65, fov = 10, zoom = .9)

rglwidget(controllers = ) # This is to get it to print in pdf

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package; using rgl.snapshot() instead

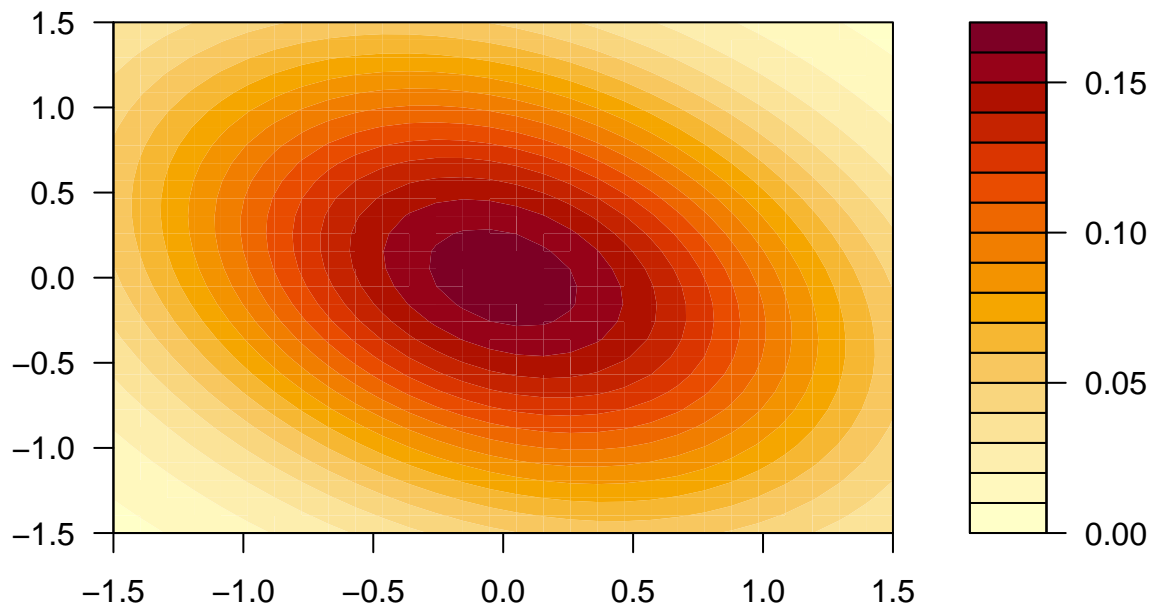
## cleared error 1285
```



(c) [3 pts] Graph the constant value contours for $f(\mathbf{x})$ for the cases (i) and (ii) in part (b). What is the shape of the constant value contours? What is the center of the constant value contours?

(d)

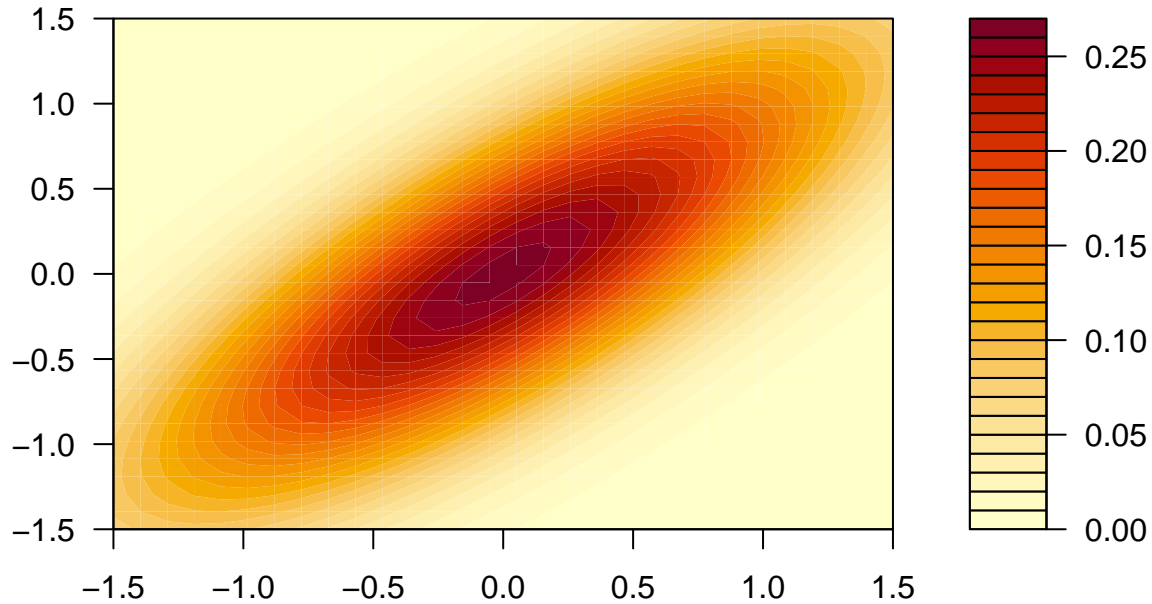
```
filled.contour(x1, x2, f_i)
```




```
#filled.contour(x1, x2, h_i)
```

(ii)

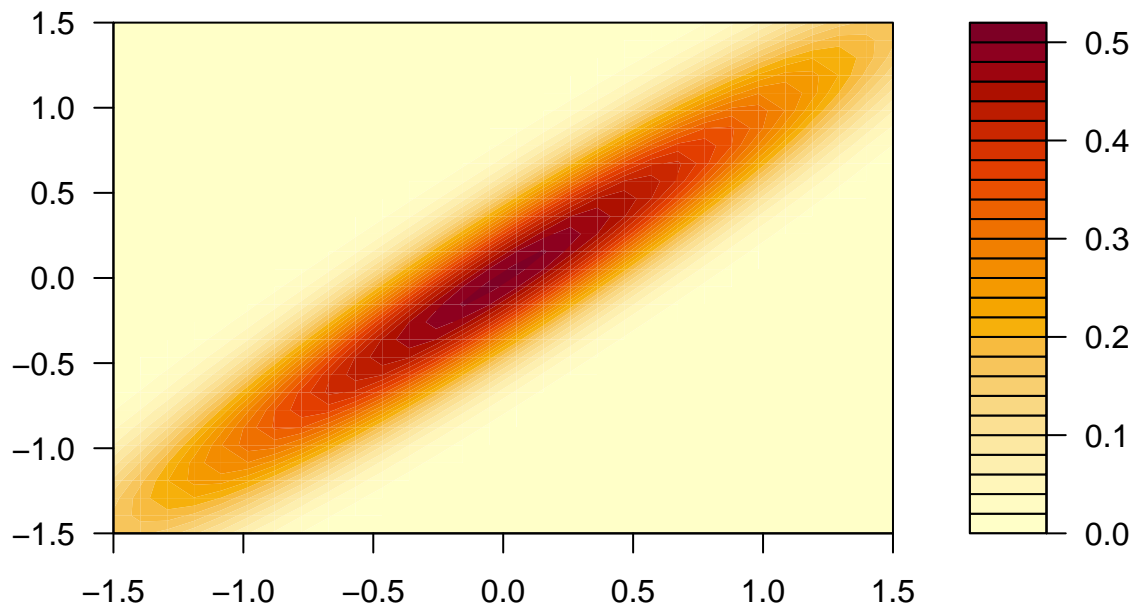
```
filled.contour(x1, x2, f_ii)
```



```
#filled.contour(x1, x2, h_iii)
```

(iii) Extra

```
filled.contour(x1, x2, f_iii)
```



```
#filled.contour(x1, x2, h_iii)
```

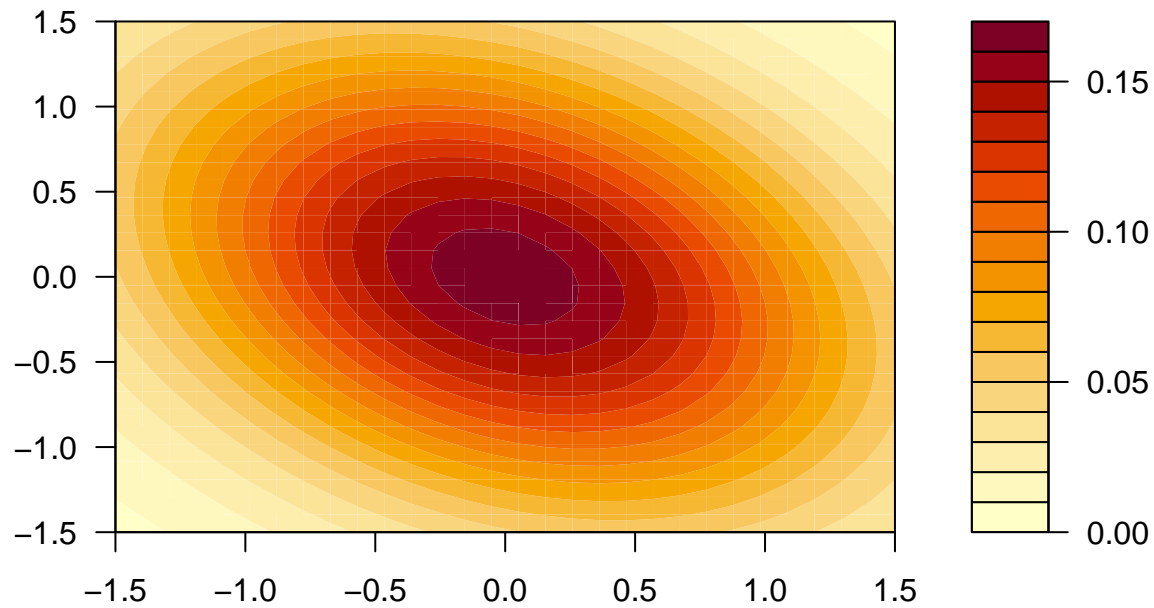
- (d) [3 pts] Compute the eigenvalues and eigenvectors for each of the covariance matrices Σ in part (b). Superimpose the eigenvectors on each of their corresponding constant value contours that you drew in part (c) and explain how the eigenvectors and eigenvalues are related to the constant value contours.

(d)(i)

```
sigma_i <- matrix(c(1,-0.3,-0.3,1), nrow=2, ncol = 2)
eigen(sigma_i)
```

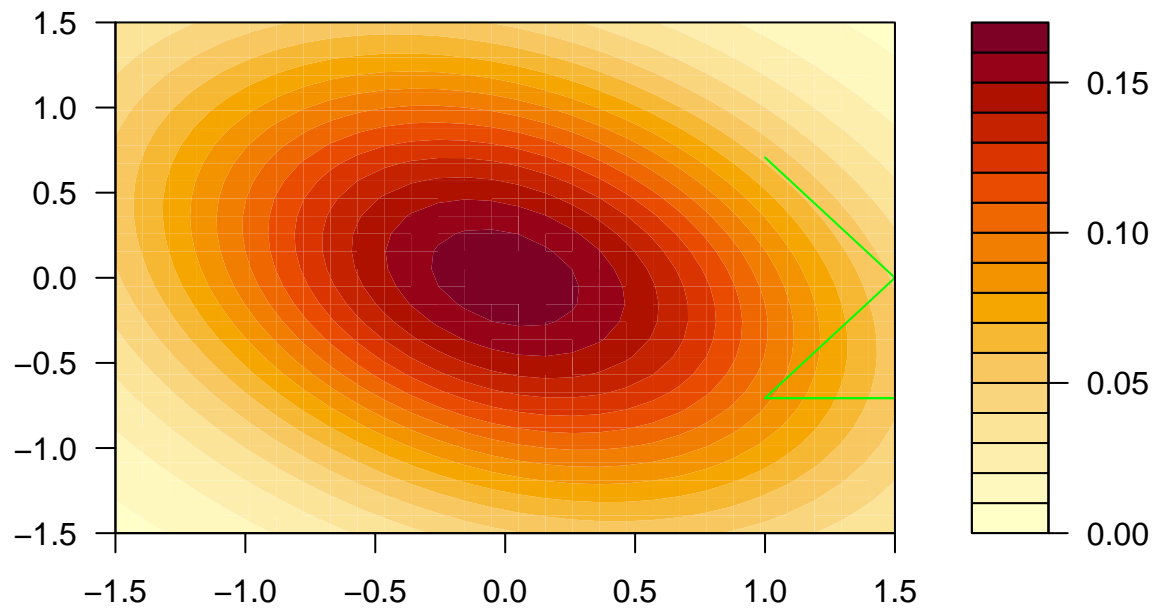
```
## eigen() decomposition
## $values
## [1] 1.3 0.7
##
## $vectors
##           [,1]      [,2]
## [1,] -0.7071068 -0.7071068
## [2,]  0.7071068 -0.7071068
```

```
filled.contour(x1, x2, f_i)
```



```
#arrows(x0=0, y0=0, x1=1, y1=0, col = "green", add = TRUE)
```

```
filled.contour(x1, x2, f_i, plot.axes = {
  axis(1)
  axis(2)
  #abline(a = 1, b = 2, col = 2)
  lines(eigen(sigma_i)$vectors[,1], col = "green")
  lines(eigen(sigma_i)$vectors[,2], col = "green")
  lines(eigen(sigma_i)$vectors[1,], col = "green")
  lines(eigen(sigma_i)$vectors[2,], col = "green")
})
```



(d)(ii)

```
sigma_ii <- matrix(c(1,0.8,0.8,1), nrow=2, ncol = 2)
eigen(sigma_ii)
```

```
## eigen() decomposition
## $values
## [1] 1.8 0.2
##
## $vectors
##      [,1]      [,2]
## [1,] 0.7071068 -0.7071068
## [2,] 0.7071068  0.7071068
```

```
filled.contour(x1, x2, f_ii)
```

