FE 621 Homework 1

Brendon Krupa

2/13/2020

Part 1

1. Data Gathering Component

```
library(quantmod)

library(jsonlite)

library(dplyr)

get_option <- function(symbol,date){

  ## Finding the end of the 3 month period
  get_3m_unix <- function(date){
    date <- as.Date(date)
    date <- date+months(3)
    date <- nth_weekday(5,date,3)
    expiry <- difftime(date,as.Date('1970-01-01'),units='secs')
    return(expiry)
  }

  #to format JSON data
  reformat_table <- function(x){
    if(is.null(x) || length(x) < 1)
      return(NULL)
    # reformat col names
    names(x) <- tolower(gsub("[[:space:]]", "", names(x)))
    # assigning the new col names
    # Vol=volume, OI=openinterest,
    d <- with(x, data.frame(Strike=strike, Last=lastprice, Chg=change,
                            Bid=bid, Ask=ask,
                            row.names=contractsymbol,
stringsAsFactors=FALSE))
    # removing commas from the data
    d[] <- lapply(d, gsub, pattern=",", replacement="", fixed=TRUE)
    d[] <- lapply(d, type.convert, as.is=TRUE)
    d
 }

  #gets calls and puts for one expiry
  get_option_data <- function(symbol,expiry){
```

```
    # url for scraping Yahoo Finance for options data using JSON
    base <- "https://query2.finance.yahoo.com/v7/finance/options/"
    url <- paste(base,symbol,"?date=",expiry,sep='')
    result <- fromJSON(url)
    #creates table
    tbl <- 
lapply(result$optionChain$result$options[[1]][,c('calls','puts')],'[[',1L)
    calls <- mapply(reformat_table,x=tbl,SIMPLIFY=F)$calls
    # assign dates
    calls['Expiry'] <- as.Date(as.POSIXct(expiry, origin="1970-01-01"), 
origin="1970-01-01")
    puts <- mapply(reformat_table,x=tbl,SIMPLIFY=F)$puts
    puts['Expiry'] <- calls['Expiry']
    return(list(calls,puts))
  }
  # calling all helper functions to pull chains for all expirations in next 3 
months
  date <- as.Date(date)
  expiry <- as.Date("2020-04-16")
  base <- "https://query2.finance.yahoo.com/v7/finance/options/"
  url <- paste(base,symbol,"?date=",expiry,sep='')
  result <- fromJSON(url)
  underlying <- result$optionChain$result$quote
  # finding all expirations for a given underlying
  available.expiries <- result$optionChain$result$expirationDates
  # filtering to those that are within the period defined above as 3 months
  expiries <- available.expiries[[1]][available.expiries[[1]] <= expiry]
  if(length(expiries)==0)
    {expiries<-available.expiries[[1]][1]}
  calls <- do.call(rbind,lapply(expiries,
                                function(x) get_option_data(symbol,x)[[1]]))
  calls['date'] <- date
  puts <- do.call(rbind,lapply(expiries,
                               function(x) get_option_data(symbol,x)[[2]]))
  puts['date'] <- date
  return(list('symbol'=symbol,'exDates'=unique(calls['Expiry']),
              'underlyingP'=underlying,
              'calls'=calls,'puts'=puts))
}
```

2. Downloading Data for AMZN, SPY, and VIX

```
# Sampling Put and Call Data for first day: Feb 10, 2020
#VIX1 <- get_option("^VIX", date = Sys.Date())
#SPY1 <- get_option("SPY", date = Sys.Date())
#AMZN1 <- get_option("AMZN", date = Sys.Date())

# Sampling Put and Call Data for second day: Feb 11, 2020
#VIX2 <- get_option("^VIX", date = Sys.Date())
#SPY2 <- get_option("SPY", date = Sys.Date())
#AMZN2 <- get_option("AMZN", date = Sys.Date())
```

Commented out because as I explained to Professor Florescu I had some issues where I lost the data on friday and had to use Bloomberg data instead. The Bloomberg data is pulled for February 13 and 14.

```
setwd("/Users/Brendon/Documents/FE 621/HW 1")
AMZNCall1 <- read.csv("AMZN Calls 2-13.csv")
AMZNCall2 <- read.csv("AMZN Calls 2-14.csv")
AMZNPut1 <- read.csv("AMZN Puts 2-13.csv")
AMZNPut2 <- read.csv("AMZN Puts 2-14.csv")
SPYCall1 <- read.csv("SPY Calls 2-13.csv")
SPYCall2 <- read.csv("SPY Calls 2-14.csv")
SPYPut1 <- read.csv("SPY Puts 2-13.csv")
SPYPut2 <- read.csv("SPY Puts 2-14.csv")
VIXCall1 <- read.csv("VIX Calls 2-13.csv")
VIXCall2 <- read.csv("VIX Calls 2-14.csv")
VIXPut1 <- read.csv("VIX Puts 2-13.csv")
VIXPut2 <- read.csv("VIX Puts 2-14.csv")
```

At the time of downloading the data on 2/14 AMZN was at 2132.27, SPY was 336.69, and the VIX was at 14.30 and the 2/13 data is taken at close with AMZN at 2149.87, SPY at 337.06, and the VIX was 14.15

3. Description of the Assets

Each of the three underlying assets are unique with AMZN being the simplest as an equity of Amazon. SPY is the symbol for the SPDR S&P 500 ETF, which is a passive investment vehicle that is designed to track the movements of the entire market. This allows investors to gain exposure to the market as a whole without having to invest in every constituent or apply portfolio optimization to develop an asset allocation strategy to design a portfolio that tracks the market. The VIX is the symbol for the CBOE Volatility Index, which measures volatility in the S&P 500 in the coming 30 days. The VIX is an important indicator of investor sentiment as analyzing the activity of derivatives on this index can prove to be powerful in measuring how investors feel about market conditions and their expectations for the future. All three symbols are major assets with a multitude of derivatives that trade off of them with varying expirations and strike prices.

4. Interest Rate

I will be using the 6 Month Treasury Bill rate of 1.52%

Part 2

5. Black-Scholes Implementation

```
r <- .0152
BSMprice <- function(S0,K,T,r,sigma,opt='c'){
  d1 <- (1/(sigma*sqrt(T)))*(log(S0/K)+(r+.5*sigma^2)*T)
  d2 <- d1 - sigma*sqrt(T)
```

```
  # If Call
  if(opt == "c"){
    return(S0*pnorm(d1)-K*exp(-r*T)*pnorm(d2))
  }
  # Put
  else{
    return(K*exp(-r*T)*pnorm(-d2)-S0*pnorm(-d1))
  }
}
BSMprice(100,105,1,r,.15)

## [1] 4.524136

BSMprice(100,105,1,r,.15,"p")

## [1] 7.940204
```

6. Bisection Method for Calculating Implied Volatility

```
bisection <- function(f,a,b,tol=10^-6){
  c <- (a+b)/2
  #while < tolerance and max iterations hasn't been reached
  while(abs(b-a)>tol){
    if(f(c)==0){return(c)}
    if(f(a)*f(c) < 0){
      b <- c
    }
    else{
      a <- c
    }
    c <- (a+b)/2
  }
  ifelse(abs(c)<.0000001,return(NA),return(c))
}
paste('Bisection:',round(bisection(function(x) cos(x),-1,2),3), "Check if
answer is within the tolerance: ", round(cos(bisection(function(x) cos(x),-
1,2)),7))

## [1] "Bisection: 1.571 Check if answer is within the tolerance:  2e-07"

# Cleaning expiration dates to calculate time to maturity
library(lubridate)

##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date

library(DataCombine)
add_year <- function(x){
```

```r
  if (year(x) < 2000){
    year(x) <- 2020
    return(as.Date(x))
  }
  return(as.Date(x))
}
temp <- as.Date(sapply(SPYCall1$Ticker, function(x) paste(substr(toString(x),
5,11), "20", sep = "")), "%m/%d/%Y")
SPYCall1$Exp <- temp
SPYCall1 <- DropNA(SPYCall1, Var = "Exp")

## 0 rows dropped from the data frame because of missing values.

SPYCall1$Exp <- as.Date(sapply(SPYCall1$Exp, add_year))
SPYCall1$Date <- as.Date("02/13/2020", '%m/%d/%Y')


temp <- as.Date(sapply(SPYCall2$Ticker, function(x) paste(substr(toString(x),
5,11), "20", sep = "")), "%m/%d/%Y")
SPYCall2$Exp <- temp
SPYCall2 <- DropNA(SPYCall2, Var = "Exp")

## 0 rows dropped from the data frame because of missing values.

SPYCall2$Exp <- as.Date(sapply(SPYCall2$Exp, add_year))
SPYCall2$Date <- as.Date("02/14/2020", '%m/%d/%Y')


temp <- as.Date(sapply(SPYPut1$Ticker, function(x) paste(substr(toString(x),
5,11), "20", sep = "")), "%m/%d/%Y")
SPYPut1$Exp <- temp
SPYPut1 <- DropNA(SPYPut1, Var = "Exp")

## 1 rows dropped from the data frame because of missing values.

SPYPut1$Exp <- as.Date(sapply(SPYPut1$Exp, add_year))
SPYPut1$Date <- as.Date("02/13/2020", '%m/%d/%Y')


temp <- as.Date(sapply(SPYPut2$Ticker, function(x) paste(substr(toString(x),
5,11), "20", sep = "")), "%m/%d/%Y")
SPYPut2$Exp <- temp
SPYPut2 <- DropNA(SPYPut2, Var = "Exp")

## 0 rows dropped from the data frame because of missing values.

SPYPut2$Exp <- as.Date(sapply(SPYPut2$Exp, add_year))
SPYPut2$Date <- as.Date("02/14/2020", '%m/%d/%Y')


temp <- as.Date(sapply(AMZNCall1$Ticker, function(x)
paste(substr(toString(x), 6,12), "20", sep = "")), "%m/%d/%Y")
AMZNCall1$Exp <- temp
AMZNCall1 <- DropNA(AMZNCall1, Var = "Exp")
```

```
## 0 rows dropped from the data frame because of missing values.
```

```
AMZNCall1$Exp <- as.Date(sapply(AMZNCall1$Exp, add_year))
AMZNCall1$Date <- as.Date("02/13/2020", '%m/%d/%Y')

temp <- as.Date(sapply(AMZNCall2$Ticker, function(x)
paste(substr(toString(x), 6,12), "20", sep = "")), "%m/%d/%Y")
AMZNCall2$Exp <- temp
AMZNCall2 <- DropNA(AMZNCall2, Var = "Exp")
```

```
## 0 rows dropped from the data frame because of missing values.
```

```
AMZNCall2$Exp <- as.Date(sapply(AMZNCall2$Exp, add_year))
AMZNCall2$Date <- as.Date("02/14/2020", '%m/%d/%Y')

temp <- as.Date(sapply(AMZNPut1$Ticker, function(x) paste(substr(toString(x),
6,12), "20", sep = "")), "%m/%d/%Y")
AMZNPut1$Exp <- temp
AMZNPut1 <- DropNA(AMZNPut1, Var = "Exp")
```

```
## 0 rows dropped from the data frame because of missing values.
```

```
AMZNPut1$Exp <- as.Date(sapply(AMZNPut1$Exp, add_year))
AMZNPut1$Date <- as.Date("02/13/2020", '%m/%d/%Y')

temp <- as.Date(sapply(AMZNPut2$Ticker, function(x) paste(substr(toString(x),
6,12), "20", sep = "")), "%m/%d/%Y")
AMZNPut2$Exp <- temp
AMZNPut2 <- DropNA(AMZNPut2, Var = "Exp")
```

```
## 0 rows dropped from the data frame because of missing values.
```

```
AMZNPut2$Exp <- as.Date(sapply(AMZNPut2$Exp, add_year))
AMZNPut2$Date <- as.Date("02/14/2020", '%m/%d/%Y')

temp <- as.Date(sapply(VIXCall1$Ticker, function(x) paste(substr(toString(x),
5,11), "20", sep = "")), "%m/%d/%Y")
VIXCall1$Exp <- temp
VIXCall1 <- DropNA(VIXCall1, Var = "Exp")
```

```
## 0 rows dropped from the data frame because of missing values.
```

```
VIXCall1$Exp <- as.Date(sapply(VIXCall1$Exp, add_year))
VIXCall1$Date <- as.Date("02/13/2020", '%m/%d/%Y')

temp <- as.Date(sapply(VIXCall2$Ticker, function(x) paste(substr(toString(x),
5,11), "20", sep = "")), "%m/%d/%Y")
VIXCall2$Exp <- temp
VIXCall2 <- DropNA(VIXCall2, Var = "Exp")
```

```
## 0 rows dropped from the data frame because of missing values.
```

```
VIXCall2$Exp <- as.Date(sapply(VIXCall2$Exp, add_year))
VIXCall2$Date <- as.Date("02/14/2020", '%m/%d/%Y')

temp <- as.Date(sapply(VIXPut1$Ticker, function(x) paste(substr(toString(x),
5,11), "20", sep = "")), "%m/%d/%Y")
VIXPut1$Exp <- temp
VIXPut1 <- DropNA(VIXPut1, Var = "Exp")

## 0 rows dropped from the data frame because of missing values.

VIXPut1$Exp <- as.Date(sapply(VIXPut1$Exp, add_year))
VIXPut1$Date <- as.Date("02/13/2020", '%m/%d/%Y')

temp <- as.Date(sapply(VIXPut2$Ticker, function(x) paste(substr(toString(x),
5,11), "20", sep = "")), "%m/%d/%Y")
VIXPut2$Exp <- temp
VIXPut2 <- DropNA(VIXPut2, Var = "Exp")

## 0 rows dropped from the data frame because of missing values.

VIXPut2$Exp <- as.Date(sapply(VIXPut2$Exp, add_year))
VIXPut2$Date <- as.Date("02/14/2020", '%m/%d/%Y')
```

Applying to DATA1 (at the money)

```
# First for the most at the money option
# AMZN Price = 2149.87, closest strike is 2150
# SPY = 337.06, closest strike is 335
AMZN2150 <- AMZNCall1[AMZNCall1$Strike==2150,1:9][4,1:9]
AMZN_C1 <- (AMZN2150[1,3] + AMZN2150[1,4])/2
AMZN_S0 <- 2149.87
AMZN_K <- AMZN2150[1,1]
AMZN_T <- as.numeric(AMZN2150[1,8] - AMZN2150[1,9])/365
AMZN_ATM <- bisection(function(x) BSMprice(AMZN_S0,AMZN_K,AMZN_T,r,x) -
AMZN_C1, 0, 1)

SPY335 <- as.vector(SPYCall1[SPYCall1$Strike==335,1:9][4,1:9])
SPY_C1 <- (SPY335[1,3] + SPY335[1,4])/2
SPY_S0 <- 337.06
SPY_K <- 335
SPY_T <- as.numeric(SPY335[1,8] - SPY335[1,9])/365
SPY_ATM <- bisection(function(x) BSMprice(SPY_S0,SPY_K,SPY_T,r,x) - SPY_C1,
0, 1)

print(paste("AMZN at the money implied volatility: ", round(AMZN_ATM*100, 2),
"%; SPY at the money implied volatility: ", round(SPY_ATM*100, 2), "%", sep =
""))

## [1] "AMZN at the money implied volatility: 23.91%; SPY at the money
implied volatility: 12.99%"
```

Applying to DATA1 (in-the-money and out-of-the-money average)

```
# AMZN in-the-money defined as S0/K > 1.05
AMZN_in <- AMZNCall1[2149.87/AMZNCall1$Strike > 1.05, c(1,3,4,8,9)]
AMZN_in$C1 <- (AMZN_in$Bid + AMZN_in$Ask)/2
AMZN_in$S0 <- 2149.87
AMZN_in$T <- as.numeric(AMZN_in$Exp - AMZN_in$Date)/365
#AMZN_in$IV <- bisection(function(x)
BSMprice(AMZN_in$S0,AMZN_in$Strike,AMZN_in$T,r,x) - AMZN_in$C1, 0, 1)
AMZN_IV_in <- c()
for (i in seq(1,nrow(AMZN_in))) {
  temp <- bisection(function(x)
BSMprice(AMZN_in$S0[i],AMZN_in$Strike[i],AMZN_in$T[i],r,x) - AMZN_in$C1[i],
0, 2)
  AMZN_IV_in <- c(AMZN_IV_in, temp)
}
AMZN_in$IV <- AMZN_IV_in
head(AMZN_in)

##   Strike     Bid     Ask        Exp       Date       C1      S0
## 1   1000 1147.90 1151.90 2020-02-14 2020-02-13 1149.900 2149.87
## 2   1280  865.90  870.85 2020-02-14 2020-02-13  868.375 2149.87
## 3   1650  495.15  501.30 2020-02-14 2020-02-13  498.225 2149.87
## 4   1675  470.85  475.95 2020-02-14 2020-02-13  473.400 2149.87
## 5   1715  430.90  436.85 2020-02-14 2020-02-13  433.875 2149.87
## 6   1720  426.00  431.80 2020-02-14 2020-02-13  428.900 2149.87
##             T IV
## 1 0.002739726  2
## 2 0.002739726  2
## 3 0.002739726  2
## 4 0.002739726  2
## 5 0.002739726  2
## 6 0.002739726  2

print("About 30% of the data maxed out the bisection function suggesting an
implied volatility greater than 200% the data frame below filters out such
options")

## [1] "About 30% of the data maxed out the bisection function suggesting an
implied volatility greater than 200% the data frame below filters out such
options"

head(AMZN_in[AMZN_in$IV < 1.999, 1:9])

##     Strike      Bid      Ask        Exp       Date       C1      S0
## 27    1985 165.1000 165.9000 2020-02-14 2020-02-13 165.5000 2149.87
## 146   1640 508.5500 512.8999 2020-02-21 2020-02-13 510.7250 2149.87
## 152   1740 410.3999 413.2000 2020-02-21 2020-02-13 411.7999 2149.87
## 153   1750 399.3501 403.2000 2020-02-21 2020-02-13 401.2750 2149.87
## 154   1760 391.3999 393.2500 2020-02-21 2020-02-13 392.3250 2149.87
## 158   1830 322.0000 323.2000 2020-02-21 2020-02-13 322.6000 2149.87
##                T        IV
```

```
## 27  0.002739726 0.7343631
## 146 0.021917808 0.7099652
## 152 0.021917808 0.6884074
## 153 0.021917808 0.6263013
## 154 0.021917808 0.6902528
## 158 0.021917808 0.5868115
```

```r
print(paste("Therefore the mean in-the-money Implied Volatility is ",
round(mean(AMZN_in[AMZN_in$IV < 1.9999, 9])*100,2), "%", sep = ""))
```

```
## [1] "Therefore the mean in-the-money Implied Volatility is 31.07%"
```

```r
# AMZN out-of-the-money defined as S0/K < .95
AMZN_out <- AMZNCall1[2149.87/AMZNCall1$Strike < .95, c(1,3,4,8,9)]
AMZN_out$C1 <- (AMZN_out$Bid + AMZN_out$Ask)/2
AMZN_out$S0 <- 2149.87
AMZN_out$T <- as.numeric(AMZN_out$Exp - AMZN_out$Date)/365
AMZN_IV_out <- c()
for (i in seq(1,nrow(AMZN_out))) {
  temp <- bisection(function(x)
BSMprice(AMZN_out$S0[i],AMZN_out$Strike[i],AMZN_out$T[i],r,x) -
AMZN_out$C1[i], 0, 2)
  AMZN_IV_out <- c(AMZN_IV_out, temp)
}
AMZN_out$IV <- AMZN_IV_out
head(AMZN_out)
```

```
##    Strike         Bid  Ask        Exp       Date    C1      S0           T
## 91   2265 0.109999955 0.41 2020-02-14 2020-02-13 0.260 2149.87 0.002739726
## 92   2270 0.100000024 0.32 2020-02-14 2020-02-13 0.210 2149.87 0.002739726
## 93   2275 0.099999964 0.38 2020-02-14 2020-02-13 0.240 2149.87 0.002739726
## 94   2280 0.099999964 0.36 2020-02-14 2020-02-13 0.230 2149.87 0.002739726
## 95   2285 0.009999998 0.40 2020-02-14 2020-02-13 0.205 2149.87 0.002739726
## 96   2290 0.100000024 0.21 2020-02-14 2020-02-13 0.155 2149.87 0.002739726
##           IV
## 91 0.4537578
## 92 0.4568963
## 93 0.4812884
## 94 0.4945769
## 95 0.5029378
## 96 0.5008912
```

```r
print(paste("The maximum Implied Volatility for AMZN out-of-the-money is 137%
so all data is included yielding a mean Implied Volatility of ",
round(mean(AMZN_out$IV)*100,2), "%", sep = ""))
```

```
## [1] "The maximum Implied Volatility for AMZN out-of-the-money is 137% so
all data is included yielding a mean Implied Volatility of 47.54%"
```

```r
# SPY in-the-money defined as S0/K > 1.05
SPY_in <- SPYCall1[337.06/SPYCall1$Strike > 1.05, c(1,3,4,8,9)]
```

```r
SPY_in$C1 <- (SPY_in$Bid + SPY_in$Ask)/2
SPY_in$S0 <- 337.06
SPY_in$T <- as.numeric(SPY_in$Exp - SPY_in$Date)/365
SPY_IV_in <- c()
for (i in seq(1,nrow(SPY_in))) {
  temp <- bisection(function(x)
BSMprice(SPY_in$S0[i],SPY_in$Strike[i],SPY_in$T[i],r,x) - SPY_in$C1[i], 0, 2)
  SPY_IV_in <- c(SPY_IV_in, temp)
}
SPY_in$IV <- SPY_IV_in
head(SPY_in)
```

```
##   Strike       Bid      Ask        Exp       Date       C1     S0
## 1     80 257.12988 257.4399 2020-02-21 2020-02-13 257.2849 337.06
## 2    200 137.17999 137.4900 2020-02-21 2020-02-13 137.3350 337.06
## 3    230 107.17999 107.5000 2020-02-21 2020-02-13 107.3400 337.06
## 4    250  87.30000  87.4500 2020-02-21 2020-02-13  87.3750 337.06
## 5    278  59.20999  59.5300 2020-02-21 2020-02-13  59.3700 337.06
## 6    280  57.20999  57.5300 2020-02-21 2020-02-13  57.3700 337.06
##            T        IV
## 1 0.02191781 1.9999995
## 2 0.02191781 1.5279059
## 3 0.02191781 1.1510463
## 4 0.02191781 0.9426503
## 5 0.02191781 0.6375699
## 6 0.02191781 0.6168199
```

```r
print(paste("Again I filtered out the values greater than 1.999; however with
this set that only represents less than 1% of the data so it is not nearly as
significant. The filtered data is below:"))
```

```
## [1] "Again I filtered out the values greater than 1.999; however with this
set that only represents less than 1% of the data so it is not nearly as
significant. The filtered data is below:"
```

```r
head(SPY_in[SPY_in$IV < 1.999,1:9])
```

```
##   Strike       Bid       Ask        Exp       Date        C1     S0
## 2    200 137.17999 137.49001 2020-02-21 2020-02-13 137.33500 337.06
## 3    230 107.17999 107.50000 2020-02-21 2020-02-13 107.34000 337.06
## 4    250  87.30000  87.45000 2020-02-21 2020-02-13  87.37500 337.06
## 5    278  59.20999  59.53000 2020-02-21 2020-02-13  59.37000 337.06
## 6    280  57.20999  57.53000 2020-02-21 2020-02-13  57.37000 337.06
## 7    286  51.23000  51.53999 2020-02-21 2020-02-13  51.38499 337.06
##            T        IV
## 2 0.02191781 1.5279059
## 3 0.02191781 1.1510463
## 4 0.02191781 0.9426503
## 5 0.02191781 0.6375699
## 6 0.02191781 0.6168199
## 7 0.02191781 0.5610404
```

```r
print(paste("This yields a mean in-the-money Implied Volatility for the SPY
of ", round(mean(SPY_in[SPY_in$IV < 1.999,1:9]$IV)*100,2), "%", sep = ""))
```

```
## [1] "This yields a mean in-the-money Implied Volatility for the SPY of
30.94%"
```

```r
# SPY out-of-the-money defined as S0/K < .95
SPY_out <- SPYCall1[337.06/SPYCall1$Strike < .95, c(1,3,4,8,9)]
SPY_out$C1 <- (SPY_out$Bid + SPY_out$Ask)/2
SPY_out$S0 <- 337.06
SPY_out$T <- as.numeric(SPY_out$Exp - SPY_out$Date)/365
SPY_IV_out <- c()
for (i in seq(1,nrow(SPY_out))) {
  temp <- bisection(function(x)
BSMprice(SPY_out$S0[i],SPY_out$Strike[i],SPY_out$T[i],r,x) - SPY_out$C1[i],
0, 2)
  SPY_IV_out <- c(SPY_IV_out, temp)
}
SPY_out$IV <- SPY_IV_out
head(SPY_out)
```

```
##     Strike  Bid  Ask        Exp       Date          C1     S0          T
## 73   355.0 0.00 0.01 2020-02-21 2020-02-13 0.005000001 337.06 0.02191781
## 127  355.0 0.02 0.03 2020-02-28 2020-02-13 0.025000001 337.06 0.04109589
## 128  360.0 0.01 0.02 2020-02-28 2020-02-13 0.015000001 337.06 0.04109589
## 174  355.0 0.05 0.06 2020-03-06 2020-02-13 0.055000000 337.06 0.06027397
## 175  356.0 0.04 0.05 2020-03-06 2020-02-13 0.045000000 337.06 0.06027397
## 176  357.5 0.03 0.04 2020-03-06 2020-02-13 0.034999998 337.06 0.06027397
##             IV
## 73  0.12467527
## 127 0.10819864
## 128 0.12591314
## 174 0.09900618
## 175 0.10070181
## 176 0.10384226
```

```r
print(paste("The mean out-of-the-money Implied Volatility for the SPY is ",
round(mean(SPY_out$IV)*100,2), "%, and I acknowledge that this value should
be greater and have checked my results several times to find the same answer.
The previous three mean implied volatilities are consistent with a volatility
smirk with the in-the-money options being greater than the at the money
options and less than the out of money options with respect to average
implied vol.", sep = ""))
```

```
## [1] "The mean out-of-the-money Implied Volatility for the SPY is 10.13%,
and I acknowledge that this value should be greater and have checked my
results several times to find the same answer. The previous three mean
implied volatilities are consistent with a volatility smirk with the in-the-
money options being greater than the at the money options and less than the
out of money options with respect to average implied vol."
```

7. Implementation of Newton/Secant Methods

```r
# Vega function for option's derivative with respect to volatility
Vega <- function(S0,K,T,r,sigma){
  d1 <- (1/(sigma*sqrt(T)))*(log(S0/K)+(r+.5*sigma^2)*T)
  return(S0*dnorm(d1)*sqrt(T))
}

#Newton implementation
newton <- function(f,df,a,tol=10^-6){
  b <- a - f(a)/df(a)
  if(is.infinite(b) | is.na(b)){
    return(NA)
  }
  while(abs(b-a)/abs(a)>tol & abs(f(a))>tol){
    a <- b
    b <- a - f(a)/df(a)
    if(is.infinite(b) | is.na(b)){
      return(NA)
    }
  }
  return(b)
}

# Testing with at the money options
AMZN_ATM <- bisection(function(x) BSMprice(AMZN_S0,AMZN_K,AMZN_T,r,x) -
AMZN_C1, 0, 1)
SPY_ATM <- bisection(function(x) BSMprice(SPY_S0,SPY_K,SPY_T,r,x) - SPY_C1,
0, 1)

AMZN_ATM2 <- newton(function(x) BSMprice(AMZN_S0,AMZN_K,AMZN_T,r,x) -
AMZN_C1, function(x) Vega(AMZN_S0,AMZN_K,AMZN_T,r,x), -.3)
SPY_ATM2 <- newton(function(x) BSMprice(SPY_S0,SPY_K,SPY_T,r,x) - SPY_C1,
function(x) Vega(SPY_S0,SPY_K,SPY_T,r,x), -.3)

print(paste("As expected, both values are the same as those yielded from the
bisection method with AMZN at ", round(AMZN_ATM2*100,2), "%, and the SPY at
", round(SPY_ATM2*100,2), "%", sep = ""))

## [1] "As expected, both values are the same as those yielded from the
bisection method with AMZN at 23.91%, and the SPY at 12.99%"
```

Timing Bisection Method vs Newton Method

```r
# Bisect AMZN in-the-money
time_start1 <- proc.time()
AMZN_IV_in <- c()
for (i in seq(1,nrow(AMZN_in))) {
  temp <- bisection(function(x)
BSMprice(AMZN_in$S0[i],AMZN_in$Strike[i],AMZN_in$T[i],r,x) - AMZN_in$C1[i],
0, 2)
```

```r
  AMZN_IV_in <- c(AMZN_IV_in, temp)
}
bisect_time <- proc.time() - time_start1
bisect_time

##    user  system elapsed
##   0.388   0.002   0.397

# Newton AMZN in-the-money
time_start2 <- proc.time()
AMZN_IV_in2 <- c()
for (i in seq(1,nrow(AMZN_in))) {
  f <- function(x)
{return(BSMprice(AMZN_in$S0[i],AMZN_in$Strike[i],AMZN_in$T[i],r,x) -
AMZN_in$C1[i])}
  df <- function(x)
{return(Vega(AMZN_in$S0[i],AMZN_in$Strike[i],AMZN_in$T[i],r,x))}
  temp <- newton(f, df, -.3)
  AMZN_IV_in2 <- c(AMZN_IV_in2, temp)
}
newton_time <- proc.time() - time_start2
newton_time

##    user  system elapsed
##   0.092   0.001   0.094

print(paste("For AMZN in-the-money the bisection method took ",
round(bisect_time[3],3), " seconds whereas the Newton method required only ",
round(newton_time[3],3), " seconds. After removing the data that yielded NA
for the Newton method, each method yielded a mean implied volatility of ",
round(mean(AMZN_IV_in2[!is.na(AMZN_IV_in2)])*100,2), "%", sep = ""))

## [1] "For AMZN in-the-money the bisection method took 0.397 seconds whereas
the Newton method required only 0.094 seconds. After removing the data that
yielded NA for the Newton method, each method yielded a mean implied
volatility of 26.72%"

# Bisect SPY in-the-money
time_start1 <- proc.time()
SPY_IV_in <- c()
for (i in seq(1,nrow(SPY_in))) {
  temp <- bisection(function(x)
BSMprice(SPY_in$S0[i],SPY_in$Strike[i],SPY_in$T[i],r,x) - SPY_in$C1[i], 0, 2)
  SPY_IV_in <- c(SPY_IV_in, temp)
}
bisect_time <- proc.time() - time_start1
bisect_time

##    user  system elapsed
##   0.334   0.002   0.341
```

```r
# Newton SPY in-the-money
time_start2 <- proc.time()
SPY_IV_in2 <- c()
for (i in seq(1,nrow(SPY_in))) {
  f <- function(x)
{return(BSMprice(SPY_in$S0[i],SPY_in$Strike[i],SPY_in$T[i],r,x) -
SPY_in$C1[i])}
  df <- function(x)
{return(Vega(SPY_in$S0[i],SPY_in$Strike[i],SPY_in$T[i],r,x))}
  temp <- newton(f, df, -.3)
  SPY_IV_in2 <- c(SPY_IV_in2, temp)
}
newton_time <- proc.time() - time_start2
newton_time
```

```
##    user  system elapsed
##   0.077   0.001   0.078
```

```r
print(paste("For the SPY in-the-money options the bisection method took ",
round(bisect_time[3],3), " seconds whereas the Newton method required only ",
round(newton_time[3],3), " seconds. After removing the data that yielded NA
for the Newton method, each method yielded a mean implied volatility of ",
round(mean(SPY_IV_in2[!is.na(SPY_IV_in2)])*100,2), "%", sep = ""))
```

```
## [1] "For the SPY in-the-money options the bisection method took 0.341
seconds whereas the Newton method required only 0.078 seconds. After removing
the data that yielded NA for the Newton method, each method yielded a mean
implied volatility of 22.66%"
```

```r
# Bisect AMZN out-of-the-money
time_start1 <- proc.time()
AMZN_IV_out <- c()
for (i in seq(1,nrow(AMZN_out))) {
  temp <- bisection(function(x)
BSMprice(AMZN_out$S0[i],AMZN_out$Strike[i],AMZN_out$T[i],r,x) -
AMZN_out$C1[i], 0, 2)
  AMZN_IV_out <- c(AMZN_IV_out, temp)
}
bisect_time <- proc.time() - time_start1
bisect_time
```

```
##    user  system elapsed
##   0.583   0.004   0.597
```

```r
# Newton AMZN in-the-money
time_start2 <- proc.time()
AMZN_IV_out2 <- c()
for (i in seq(1,nrow(AMZN_out))) {
  f <- function(x)
{return(BSMprice(AMZN_out$S0[i],AMZN_out$Strike[i],AMZN_out$T[i],r,x) -
AMZN_out$C1[i])}
```

```
  df <- function(x)
{return(Vega(AMZN_out$S0[i],AMZN_out$Strike[i],AMZN_out$T[i],r,x))}
  temp <- newton(f, df, -.3)
  AMZN_IV_out2 <- c(AMZN_IV_out2, temp)
}
newton_time <- proc.time() - time_start2
newton_time
```

```
##    user  system elapsed
##   0.110   0.001   0.114
```

```
print(paste("For AMZN out-of-the-money options the bisection method took ",
round(bisect_time[3],3), " seconds whereas the Newton method required only ",
round(newton_time[3],3), " seconds. After removing the data that yielded NA
for the Newton method, each method yielded a mean implied volatility of ",
round(mean(AMZN_IV_out2[!is.na(AMZN_IV_out2)])*100,2), "%", sep = ""))
```

```
## [1] "For AMZN out-of-the-money options the bisection method took 0.597
seconds whereas the Newton method required only 0.114 seconds. After removing
the data that yielded NA for the Newton method, each method yielded a mean
implied volatility of 28.75%"
```

```
# Bisect SPY out-of-the-money
time_start1 <- proc.time()
SPY_IV_out <- c()
for (i in seq(1,nrow(SPY_out))) {
  temp <- bisection(function(x)
BSMprice(SPY_out$S0[i],SPY_out$Strike[i],SPY_out$T[i],r,x) - SPY_out$C1[i],
0, 2)
  SPY_IV_out <- c(SPY_IV_out, temp)
}
bisect_time <- proc.time() - time_start1
bisect_time
```

```
##    user  system elapsed
##   0.171   0.001   0.177
```

```
# Newton SPY in-the-money
time_start2 <- proc.time()
SPY_IV_out2 <- c()
for (i in seq(1,nrow(SPY_out))) {
  f <- function(x)
{return(BSMprice(SPY_out$S0[i],SPY_out$Strike[i],SPY_out$T[i],r,x) -
SPY_out$C1[i])}
  df <- function(x)
{return(Vega(SPY_out$S0[i],SPY_out$Strike[i],SPY_out$T[i],r,x))}
  temp <- newton(f, df, -.3)
  SPY_IV_out2 <- c(SPY_IV_out2, temp)
}
newton_time <- proc.time() - time_start2
newton_time
```

```
##    user  system elapsed
##   0.070   0.000   0.071
```

```r
print(paste("For SPY out-of-the-money options the bisection method took ",
round(bisect_time[3],3), " seconds whereas the Newton method required only ",
round(newton_time[3],3), " seconds. After removing the data that yielded NA
for the Newton method, each method yielded a mean implied volatility of ",
round(mean(SPY_IV_out2[!is.na(SPY_IV_out2)])*100,2), "%", sep = ""))
```

```
## [1] "For SPY out-of-the-money options the bisection method took 0.177
seconds whereas the Newton method required only 0.071 seconds. After removing
the data that yielded NA for the Newton method, each method yielded a mean
implied volatility of 10.01%"
```

8. Summary Table of Implied Volatility for Each Option Type, Maturity, and Stock

```r
#aggregate(SPY_in[, 9],list(SPY_in$Exp), mean)
AMZNCall1$C1 <- (AMZNCall1$Bid + AMZNCall1$Ask)/2
AMZNCall1$S0 <- 2149.87
AMZNCall1$T <- as.numeric(AMZNCall1$Exp - AMZNCall1$Date)/365
AMZN_IV <- c()
for (i in seq(1,nrow(AMZNCall1))) {
  temp <- bisection(function(x)
BSMprice(AMZNCall1$S0[i],AMZNCall1$Strike[i],AMZNCall1$T[i],r,x) -
AMZNCall1$C1[i], 0, 2)
  AMZN_IV <- c(AMZN_IV, temp)
}
AMZNCall1$IV <- AMZN_IV
temp <- AMZNCall1[AMZNCall1$IV < 1.999, 1:13]
AMZNCall1table <- aggregate(temp[,13], list(temp$Exp), mean)
colnames(AMZNCall1table) <- c("Maturity", "Implied Vol (%)")
AMZNCall1table$`Implied Vol` <- round(100*AMZNCall1table$`Implied Vol`,2)

#aggregate(SPY_in[, 9],list(SPY_in$Exp), mean)
AMZNPut1$C1 <- (AMZNPut1$Bid + AMZNPut1$Ask)/2
AMZNPut1$S0 <- 2149.87
AMZNPut1$T <- as.numeric(AMZNPut1$Exp - AMZNPut1$Date)/365
AMZN_IV <- c()
for (i in seq(1,nrow(AMZNPut1))) {
  temp <- bisection(function(x)
BSMprice(AMZNPut1$S0[i],AMZNPut1$Strike[i],AMZNPut1$T[i],r,x,"p") -
AMZNPut1$C1[i], 0, 2)
  AMZN_IV <- c(AMZN_IV, temp)
}
AMZNPut1$IV <- AMZN_IV
temp <- AMZNPut1[AMZNPut1$IV < 1.999, 1:13]
AMZNPut1table <- aggregate(temp[,13], list(temp$Exp), mean)
colnames(AMZNPut1table) <- c("Maturity", "Implied Vol (%)")
AMZNPut1table$`Implied Vol` <- round(100*AMZNPut1table$`Implied Vol`,2)
```

```r
SPYCall1$C1 <- (SPYCall1$Bid + SPYCall1$Ask)/2
SPYCall1$S0 <- 337.06
SPYCall1$T <- as.numeric(SPYCall1$Exp - SPYCall1$Date)/365
SPY_IV <- c()
for (i in seq(1,nrow(SPYCall1))) {
  temp <- bisection(function(x)
BSMprice(SPYCall1$S0[i],SPYCall1$Strike[i],SPYCall1$T[i],r,x) -
SPYCall1$C1[i], 0, 2)
  SPY_IV <- c(SPY_IV, temp)
}
SPYCall1$IV <- SPY_IV
temp <- SPYCall1[SPYCall1$IV < 1.999, 1:13]
SPYCall1table <- aggregate(temp[,13], list(temp$Exp), mean)
colnames(SPYCall1table) <- c("Maturity", "Implied Vol (%)")
SPYCall1table$`Implied Vol` <- round(100*SPYCall1table$`Implied Vol`,2)
```

Average Volatilities

```r
mean_table <- data.frame(matrix(nrow = 3, ncol = 2))
mean_table[1,1] <- round(mean(SPY_in[SPY_in$IV < 1.999,1:9]$IV)*100,2)
mean_table[2,1] <- round(SPY_ATM*100, 2)
mean_table[3,1] <- round(mean(SPY_out$IV)*100,2)
mean_table[1,2] <- round(mean(AMZN_in[AMZN_in$IV < 1.9999, 9])*100,2)
mean_table[2,2] <- round(AMZN_ATM*100, 2)
mean_table[3,2] <- round(mean(AMZN_out$IV)*100,2)
colnames(mean_table) <- c("SPY", "AMZN")
rownames(mean_table) <- c("In-The-Money (%)", "At-The-Money (%)", "Out-Of-
Money (%)")
mean_table
```

```
##                    SPY  AMZN
## In-The-Money (%) 30.94 31.07
## At-The-Money (%) 12.99 23.91
## Out-Of-Money (%) 10.13 47.54
```

Amazon clearly illustrates the volatility smirk with the out-of-money options having a far higher implied volatility than the at-the-money options and significantly greater than the in-the-money options as well. As I mentioned earlier the out-of-money calculation for the SPY seems to be off; however, the in-the-money calculation is more reasonable and the at-the-money implied volatility is close to the value of the VIX.

```r
print(paste("Amazon Calls Summary Table"))
```

```
## [1] "Amazon Calls Summary Table"
```

```r
AMZNCall1table[,c(1,3)]
```

```
##     Maturity Implied Vol
## 1 2020-02-14       63.47
## 2 2020-02-21       38.07
## 3 2020-02-28       31.18
## 4 2020-03-06       29.76
```

```
## 5 2020-03-13       28.18
## 6 2020-03-20       28.36
```

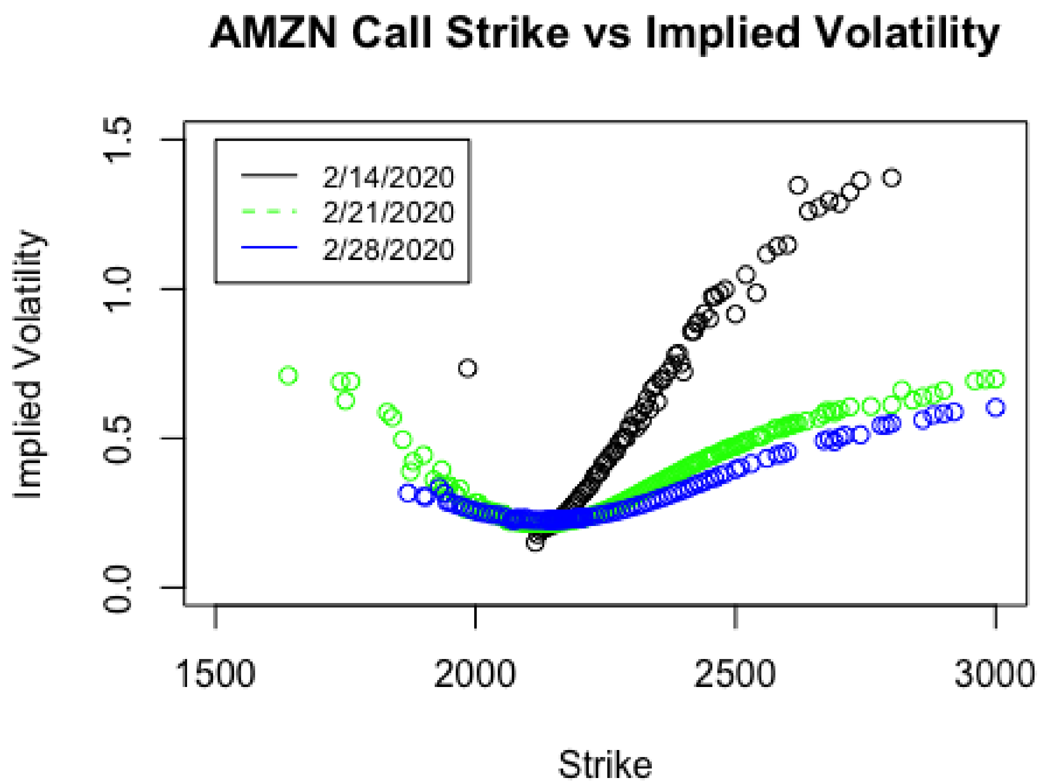
The implied volatility decreases as the maturity increases because the option has more time for the price of the underlying to reach the strike price in order for the option to become at or in the money. As I alluded to above the volatility smirk is evident with the data for Amazon where in the money options have a greater implied volatility than at the money options and the out of the money options yield the greatest implied volatility. The remaining tables for the other stocks and option types are included in the appendix.

9. Put-Call Parity Calculations

```
#P-C parity is C + Ke^-rT = S + P
#SPYCall1$Put <- SPYCall1$C1 + SPYCall1$Strike*exp(-r*SPYCall1$T) -
SPYCall1$S0
#SPYPut1$Call <- SPYPut1$C1 + SPYPut1$S0 - SPYPut1$Strike*exp(-r*SPYPut1$T)

#AMZNCall1$Put <- AMZNCall1$C1 + AMZNCall1$Strike*exp(-r*AMZNCall1$T) -
AMZNCall1$S0
#AMZNPut1$Call <- AMZNPut1$C1 + AMZNPut1$S0 - AMZNPut1$Strike*exp(-
r*AMZNPut1$T)
```

10. Implied Volatility vs Strike Price

```
temp1 <- AMZNCall1[AMZNCall1$Exp == as.Date("2020-02-14"), 1:13]
temp2 <- AMZNCall1[AMZNCall1$Exp == as.Date("2020-02-21"), 1:13]
temp3 <- AMZNCall1[AMZNCall1$Exp == as.Date("2020-02-28"), 1:13]
plot(temp1$Strike, temp1$IV, xlab = "Strike", ylab = "Implied Volatility",
main = "AMZN Call Strike vs Implied Volatility", ylim = c(0,1.5), xlim =
c(1500, 3000))
lines(temp2$Strike, temp2$IV, type = "p", col = c("green"))
lines(temp3$Strike, temp3$IV, type = "p", col = c("blue"))
legend(1500, 1.5,legend = c("2/14/2020", "2/21/2020", "2/28/2020"), col =
c("black", "green", "blue"), lty=1:2, cex=0.8)
```



The general pattern of the volatility smirk can be seen although and it is interesting to see how the shape differs for each maturity. The SPY graphs are included in the appendix.

11. Greeks

```
theoretical_greeks <- function(S0,K,tau,r,sigma,type='c'){
  N <- function(x){pnorm(x)}
  d1 <- (1/(sigma*sqrt(tau)))*(log(S0/K)+(r+.5*sigma**2)*tau)
  delta <- ifelse(type=='c',N(d1),N(d1)-1)
  gamma <- 1/(S0*sigma*sqrt(tau))*exp(-d1^2/2)/(sqrt(2*pi))
  vega <- S0*sqrt(tau)*exp(-d1^2/2)/sqrt(2*pi)
  x = data.frame(Delta=delta,Gamma=gamma,Vega=vega)
  return(x)
}
numerical_greeks <- function(S0,K,tau,r,sigma,type='c',e=10^-4){
  delta <- (BSMprice(S0+e,K,tau,r,sigma)-BSMprice(S0-e,K,tau,r,sigma))/(2*e)
  gamma <- (BSMprice(S0+e,K,tau,r,sigma)-2*BSMprice(S0,K,tau,r,sigma)+
            BSMprice(S0-e,K,tau,r,sigma))/e^2
  vega <- (BSMprice(S0,K,tau,r,sigma+e)-BSMprice(S0,K,tau,r,sigma-e))/(2*e)
  x = data.frame(Delta=delta,Gamma=gamma,Vega=vega)
  return(x)
}
numerical_greeks(100,100,1,0.05,0.3,'c')
```

```
##       Delta      Gamma     Vega
## 1 0.6242517 0.01264979 37.94329
```

```r
theoretical_greeks(100,100,1,0.05,0.3,'c')
```

```
##       Delta      Gamma     Vega
## 1 0.6242517 0.01264776 37.94329
```

```r
num <-
numerical_greeks(AMZNCall1$S0,AMZNCall1$Strike,AMZNCall1$T,r,AMZNCall1$IV)
theo <-
theoretical_greeks(AMZNCall1$S0,AMZNCall1$Strike,AMZNCall1$T,r,AMZNCall1$IV)
num[80:90,1:3]
```

```
##         Delta        Gamma      Vega
## 80 0.05166667 0.0029316993 11.912774
## 81 0.04334856 0.0024641622 10.349594
## 82 0.03661931 0.0020975222  9.023430
## 83 0.03375716 0.0018744117  8.440607
## 84 0.03203902 0.0017081447  8.084887
## 85 0.02719211 0.0014502177  7.055746
## 86 0.02205571 0.0012057910  5.918775
## 87 0.02319266 0.0011880275  6.174959
## 88 0.01977186 0.0010125234  5.395675
## 89 0.01734363 0.0008867573  4.826026
## 90 0.01654460 0.0008228085  4.635280
```

```r
theo[80:90,1:3]
```

```
##    Delta        Gamma      Vega
## 80     1 0.0029271715 11.912774
## 81     1 0.0024714229 10.349594
## 82     1 0.0020945290  9.023429
## 83     1 0.0018687254  8.440607
## 84     1 0.0017018866  8.084887
## 85     1 0.0014538867  7.055746
## 86     1 0.0012064742  5.918775
## 87     1 0.0011817186  6.174959
## 88     1 0.0010148353  5.395675
## 89     1 0.0008879334  4.826026
## 90     1 0.0008224522  4.635279
```

The Gamma and Vega values are very similar for each method; however it seems that my delta calculation is off for the theoretical method causing the discrepency. The full table is featured in the appendix.

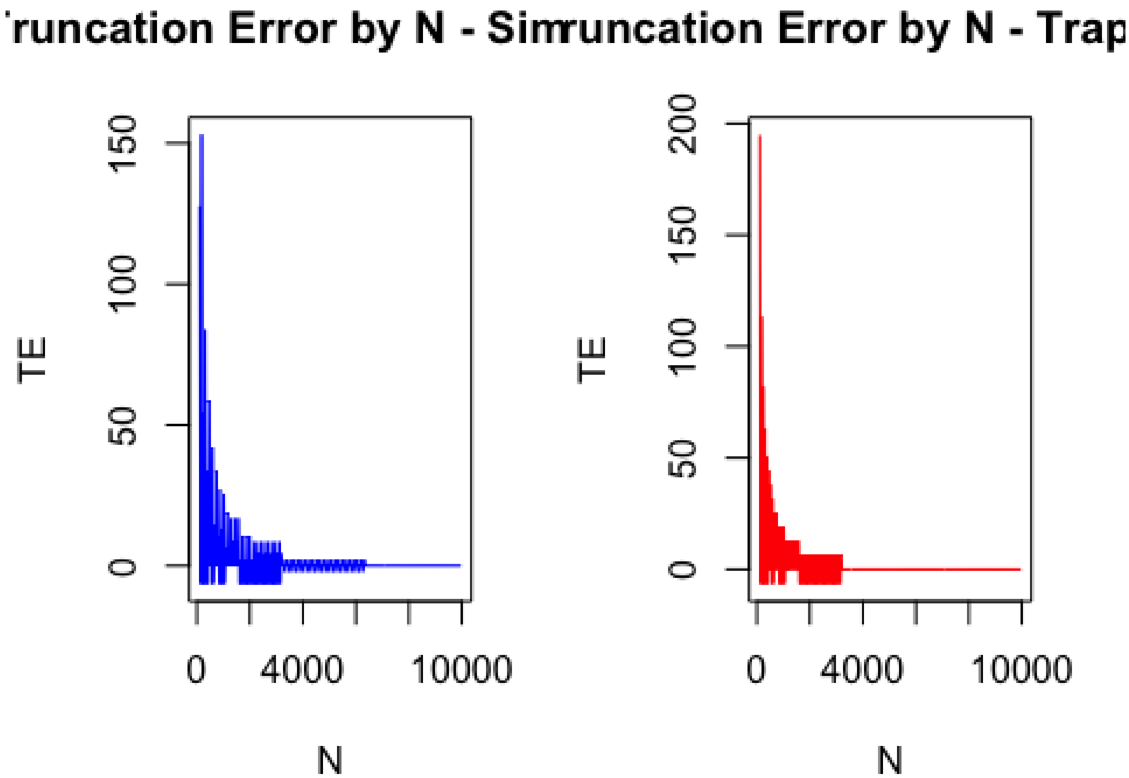
Part 2

1. Trapezoidal and Simpson's Quadrature Rules for Approximation

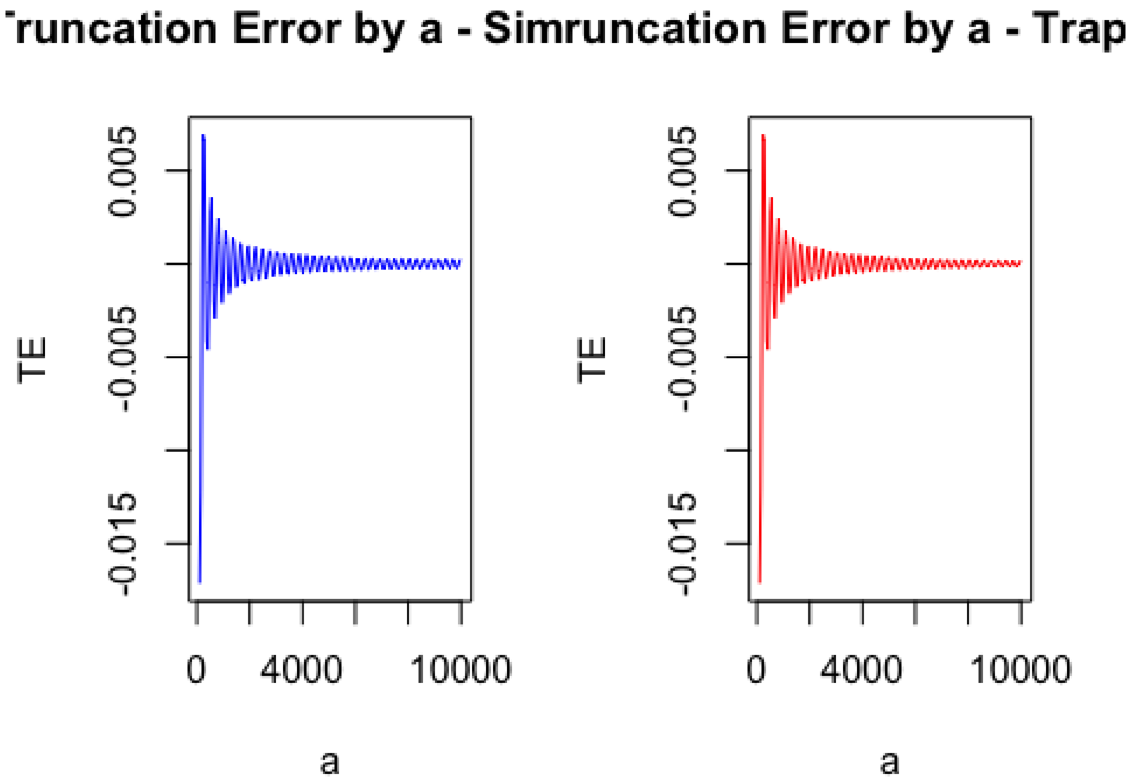
```
fx <- function(x){
  fx <- ifelse(x==0,1,sin(x)/x)
  return(fx)
}
simpson <- function(f,a,b,n){
  h <- (b-a)/(n)
  x <- seq(a,b,by=h)
  y <- f(x)
  s <- y[1]+y[n+1]+4*sum(y[seq(2,n,by=2)])+
    2*sum(y[seq(3,n-1,by=2)])
  s <- (h/3)*s
  return(s)
}
trapezoidal <- function(f,a,b,n){
  h <- (b-a)/n
  x <- seq(a,b,by=h)
  y <- f(x)
  s <- (h/2)*(y[1]+2*sum(y[2:n])+y[n+1])
  return(s)
}
trapezoidal(fx,-10^6,10^6,n=5000000)

## [1] 3.141591

simpson(fx,-10^6,10^6,n=5000000)

## [1] 3.141591
```

2. Computing Truncation Errror

```
calc_TE <- function(func,fx,a,N){
  val <- func(fx,-a,a,N)
  te <- val-pi
  return(te)
}
#Fixing a
par(mfrow=c(1,2))
a <- 1e4
N <- seq(100,1e4,by=35)
TE_simp <- sapply(N,calc_TE,a=a,func=simpson,fx=fx)
TE_trap <- sapply(N,calc_TE,a=a,func=trapezoidal,fx=fx)
plot(N,TE_simp,type='l',col='blue',xlab='N',ylab = 'TE',
     main = 'Truncation Error by N - Simpson')
plot(N,TE_trap,type='l',col='red',xlab='N',ylab = 'TE',
     main = 'Truncation Error by N - Trapezoid')
```




```
#Fixing N
N <- 1e4
a <- seq(100,1e4,by=45)
TE_simp <- sapply(a,calc_TE,N=N,func=simpson,fx=fx)
TE_trap <- sapply(a,calc_TE,N=N,func=trapezoidal,fx=fx)
plot(a,TE_simp,type='l',col='blue',xlab='a',ylab = 'TE',
     main = 'Truncation Error by a - Simpson')
plot(a,TE_trap,type='l',col='red',xlab='a',ylab = 'TE',
     main = 'Truncation Error by a - Trapezoid')
```



From the first plots, we see that the Trapezoidal rule converges before the Simpson rule, though the errors for the Simpson rule appear to be lower for each N. We also see that as N increases, the Truncation Error approaches or equals zero, which is expected, as smaller partitions lead to increasingly miniscule error.

Holding N constant, we see that the choice of a causes the Truncation Error to fluctuate around zero. As a increases, we see that the magnitude of these flucations become less signficant until the error seems to converge to zero. The Truncation Error by a plots appear identical between the two methods. It is clear that as the boundaries of integration increase, the approximation becomes more accurate.

3. Evaluating Number of Steps Required to Converge

```
steps <- function(f,fx,a,b,tol=10^-4){
  n <- 5
  error <-100
  v1 <- f(fx,a,b,n-1)
  while(error>tol){
    v2 <- f(fx,a,b,n)
    error <- abs(v2-v1)
    v1 <- v2
    n <- n+1
  }
```

```
  l <- list(value=v2,steps=n-5)
  return(l)
}
print(paste("Trapezoidal:"))

## [1] "Trapezoidal:"

steps(trapezoidal,fx,-1e3,1e3)

## $value
## [1] 3.146739
##
## $steps
## [1] 377

print(paste("Simpson:"))

## [1] "Simpson:"

steps(simpson,fx,-1e3,1e3)

## $value
## [1] 3.140543
##
## $steps
## [1] 1541
```

While the Simpson method is more accurate, it takes far more steps to reach that accuracy. Therein lies an important tradeoff between speed and accuracy.

Part 4

1. Solving Integral for f1 and f2

```
f1 <- function(x,y) {return(x*y)}
f2 <- function(x,y) {return(exp(x+y))}

print(paste("Anlytical Solution to f1"))

## [1] "Anlytical Solution to f1"

integrate(function(y) {
   sapply(y, function(y) {
     integrate(function(x) f1(x,y), 0, 3)$value
   })
 }, 0, 1)

## 2.25 with absolute error < 2.5e-14

print(paste("Anlytical Solution to f2"))

## [1] "Anlytical Solution to f2"
```

```
integrate(function(y) {
   sapply(y, function(y) {
     integrate(function(x) f2(x,y), 0, 3)$value
   })
 }, 0, 1)

## 32.79433 with absolute error < 3.6e-13
```

2. Numerical Solution

```
deltaX <- 0.001
deltaY <- 0.003
x0 <- 0
xn1 <- 1
y0 <- 0
ym1 <- 3

x<- seq(x0,xn1,deltaX)
y<- seq(y0,ym1,deltaY)
s <- 0
for(i in seq(1,length(x))-1){
  s <- s + f1(x[i],y[i]) + f1(x[i],y[i+1]) + f1(x[i+1],y[i]) +
f1(x[i+1],y[i+1]) + 2*(f1((x[i]+x[i+1])/2,y[i]) + f1((x[i]+x[i+1])/2,y[i+1])
+ f1(x[i],(y[i] + y[i+1])/2) + f1(x[i+1],(y[i] + y[i+1])/2)) + 4*f1((x[i] +
x[i+1])/2, (y[i]+y[i+1])/2)
}
s*deltaX*deltaY/16

## numeric(0)
```

Appendix

```
print(paste("Amazon Calls Summary Table"))

## [1] "Amazon Calls Summary Table"

AMZNCall1table[,c(1,3)]

##     Maturity Implied Vol
## 1 2020-02-14       63.47
## 2 2020-02-21       38.07
## 3 2020-02-28       31.18
## 4 2020-03-06       29.76
## 5 2020-03-13       28.18
## 6 2020-03-20       28.36

print(paste("Amazon Puts Summary Table"))

## [1] "Amazon Puts Summary Table"

AMZNPut1table[,c(1,3)]
```

```
##     Maturity Implied Vol
## 1 2020-02-14       78.08
## 2 2020-02-21       43.60
## 3 2020-02-28       31.46
## 4 2020-03-06       28.30
## 5 2020-03-13       27.83
## 6 2020-03-20       29.76
## 7 2020-03-27       26.35
## 8 2020-04-03       23.86
## 9 2020-04-17       26.69
```

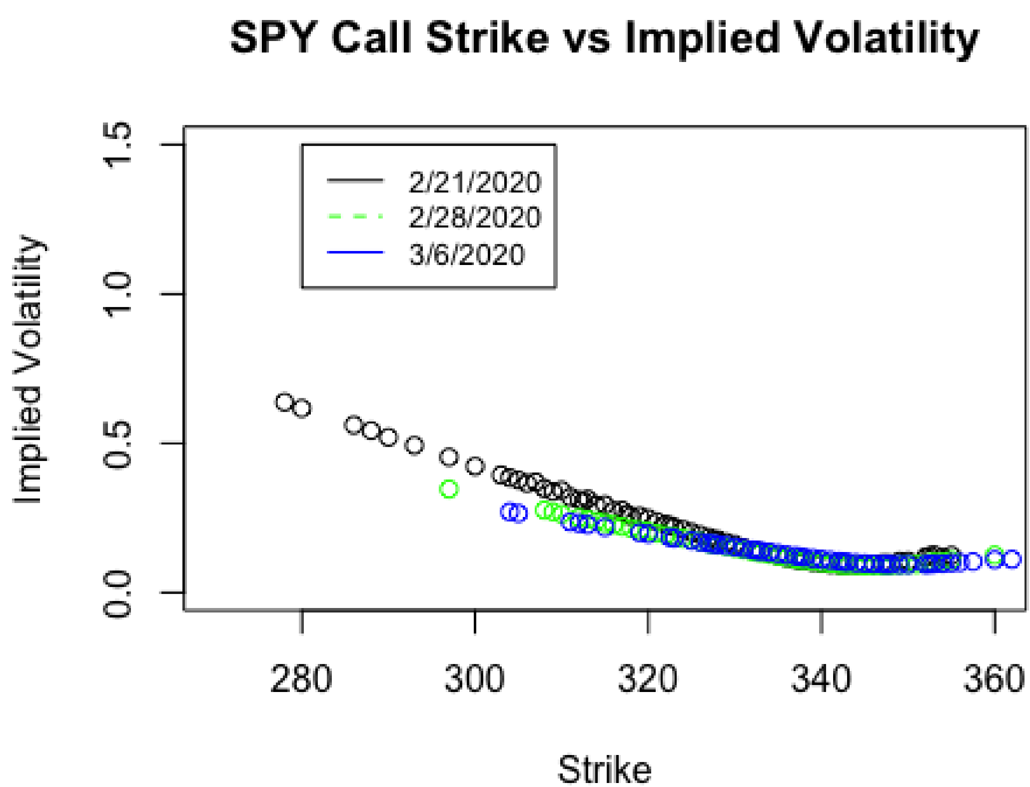
```
print(paste("SPY Calls Summary Table"))
```

```
## [1] "SPY Calls Summary Table"
```

```
SPYCall1table
```

```
##     Maturity Implied Vol (%) Implied Vol
## 1 2020-02-21       0.2751647       27.52
## 2 2020-02-28       0.1522963       15.23
## 3 2020-03-06       0.1395745       13.96
## 4 2020-03-13       0.1284705       12.85
## 5 2020-03-20       0.2206953       22.07
## 6 2020-03-27       0.1125911       11.26
## 7 2020-04-03       0.1026095       10.26
## 8 2020-04-17       0.1239359       12.39
```

```
temp1 <- SPYCall1[SPYCall1$Exp == as.Date("2020-03-06"), 1:13]
temp2 <- SPYCall1[SPYCall1$Exp == as.Date("2020-02-21"), 1:13]
temp3 <- SPYCall1[SPYCall1$Exp == as.Date("2020-02-28"), 1:13]
plot(temp2$Strike, temp2$IV, xlab = "Strike", ylab = "Implied Volatility",
main = "SPY Call Strike vs Implied Volatility", xlim = c(270, 360), ylim =
c(0,1.5))
lines(temp3$Strike, temp3$IV, type = "p", col = c("green"))
lines(temp1$Strike, temp1$IV, type = "p", col = c("blue"))
legend(280, 1.5,legend = c("2/21/2020", "2/28/2020", "3/6/2020"), col =
c("black", "green", "blue"), lty=1:2, cex=0.8)
```




```
num

##             Delta        Gamma         Vega
## 1   1.0000000020 0.000000e+00 0.000000e+00
## 2   0.9999997224 0.000000e+00 1.622266e-04
## 3   0.9950690674 9.094947e-05 1.607108e+00
## 4   0.9925948791 2.273737e-05 2.304410e+00
## 5   0.9865010520 1.818989e-04 3.891532e+00
## 6   0.9855092810 1.591616e-04 4.136780e+00
## 7   0.9499022553 5.002221e-04 1.162387e+01
## 8   0.9381596897 5.911716e-04 1.372522e+01
## 9   0.9332768354 5.911716e-04 1.456034e+01
## 10  0.9315891623 6.139089e-04 1.484410e+01
## 11  0.9245323486 6.366463e-04 1.600481e+01
## 12  0.9169761734 6.821210e-04 1.720399e+01
## 13  0.9130072283 7.048584e-04 1.781684e+01
## 14  0.9089093783 6.821210e-04 1.843785e+01
## 15  0.8912090448 8.185452e-04 2.099221e+01
## 16  0.8765404880 9.322321e-04 2.296450e+01
## 17  0.8739788279 9.322321e-04 2.329647e+01
## 18  0.8660934054 9.549694e-04 2.429617e+01
## 19  0.8606697054 9.777068e-04 2.496488e+01
## 20  0.8579079633 1.023182e-03 2.529963e+01
```

```
## 21  0.8494240012 1.023182e-03 2.630430e+01
## 22  0.8376518520 1.068656e-03 2.764121e+01
## 23  0.8253610952 1.159606e-03 2.896928e+01
## 24  0.8190244171 1.227818e-03 2.962799e+01
## 25  0.8125622207 1.182343e-03 3.028207e+01
## 26  0.7992683231 1.273293e-03 3.157312e+01
## 27  0.9819613740 5.229595e-04 4.990574e+00
## 28  0.7854950752 1.250555e-03 3.283596e+01
## 29  0.7784342733 1.318767e-03 3.345480e+01
## 30  0.7712606407 1.409717e-03 3.406421e+01
## 31  0.7639768182 1.318767e-03 3.466341e+01
## 32  0.7565855947 1.386979e-03 3.525163e+01
## 33  0.7490899031 1.432454e-03 3.582812e+01
## 34  0.7414927995 1.455192e-03 3.639214e+01
## 35  0.7337974819 1.477929e-03 3.694299e+01
## 36  0.7260072653 1.500666e-03 3.747995e+01
## 37  0.7181255887 1.500666e-03 3.800235e+01
## 38  0.7141515573 1.568878e-03 3.825789e+01
## 39  0.7101560004 1.455192e-03 3.850953e+01
## 40  0.7021021634 1.546141e-03 3.900085e+01
## 41  0.6980448234 1.546141e-03 3.924038e+01
## 42  0.6939678383 1.591616e-03 3.947571e+01
## 43  0.6898716981 1.568878e-03 3.970678e+01
## 44  0.6857568872 1.659828e-03 3.993351e+01
## 45  0.6816239045 1.568878e-03 4.015584e+01
## 46  0.6774732537 1.637090e-03 4.037370e+01
## 47  0.6733054374 1.591616e-03 4.058703e+01
## 48  0.6691209728 1.637090e-03 4.079575e+01
## 49  0.6649203692 1.591616e-03 4.099981e+01
## 50  0.6607041519 1.682565e-03 4.119915e+01
## 51  0.6564728392 1.637090e-03 4.139371e+01
## 52  0.6522269678 1.637090e-03 4.158344e+01
## 53  0.6479670628 1.682565e-03 4.176828e+01
## 54  0.6436936621 1.614353e-03 4.194817e+01
## 55  0.6351085381 1.728040e-03 4.229293e+01
## 56  0.6264759384 1.750777e-03 4.261734e+01
## 57  0.6178002536 1.728040e-03 4.292107e+01
## 58  0.6090859097 1.682565e-03 4.320379e+01
## 59  0.6003373539 1.750777e-03 4.346524e+01
## 60  0.5915590646 1.773515e-03 4.370515e+01
## 61  0.9811666723 2.728484e-03 5.177229e+00
## 62  0.9351888491 6.343726e-03 1.423586e+01
## 63  0.8816848162 9.322321e-03 2.228685e+01
## 64  0.8167491455 1.186891e-02 2.986030e+01
## 65  0.7506490351 1.418812e-02 3.570985e+01
## 66  0.6704142811 1.573426e-02 4.073182e+01
## 67  0.5847352611 1.639364e-02 4.387623e+01
## 68  0.5015147940 1.687113e-02 4.489239e+01
## 69  0.4221292539 1.570015e-02 4.403476e+01
## 70  0.3501473185 1.447233e-02 4.168714e+01
```

```
## 71  0.2881204585 1.285798e-02 3.840163e+01
## 72  0.2352666067 1.118110e-02 3.460207e+01
## 73  0.1910952815 9.623591e-03 3.064451e+01
## 74  0.1561276548 8.151346e-03 2.694288e+01
## 75  0.1290947773 6.923528e-03 2.369007e+01
## 76  0.1057979404 5.780976e-03 2.057421e+01
## 77  0.0887390550 4.899903e-03 1.808291e+01
## 78  0.0736842338 4.098410e-03 1.571531e+01
## 79  0.0636396791 3.524292e-03 1.403546e+01
## 80  0.0516666689 2.931699e-03 1.191277e+01
## 81  0.0433485589 2.464162e-03 1.034959e+01
## 82  0.0366193093 2.097522e-03 9.023430e+00
## 83  0.0337571618 1.874412e-03 8.440607e+00
## 84  0.0320390173 1.708145e-03 8.084887e+00
## 85  0.0271921121 1.450218e-03 7.055746e+00
## 86  0.0220557070 1.205791e-03 5.918775e+00
## 87  0.0231926582 1.188027e-03 6.174959e+00
## 88  0.0197718588 1.012523e-03 5.395675e+00
## 89  0.0173436290 8.867573e-04 4.826026e+00
## 90  0.0165446004 8.228085e-04 4.635280e+00
## 91  0.0145244696 7.247536e-04 4.145072e+00
## 92  0.0119147204 6.036061e-04 3.493045e+00
## 93  0.0128263181 6.107115e-04 3.723391e+00
## 94  0.0120480123 5.620393e-04 3.526909e+00
## 95  0.0106993012 4.977352e-04 3.181201e+00
## 96  0.0083384288 4.032330e-04 2.558043e+00
## 97  0.0107193218 4.682477e-04 3.186384e+00
## 98  0.0087158261 3.868905e-04 2.659354e+00
## 99  0.0103781385 4.263256e-04 3.097840e+00
## 100 0.0054980770 2.602363e-04 1.770578e+00
## 101 0.0079219700 3.318235e-04 2.445427e+00
## 102 0.0048346218 2.239986e-04 1.579102e+00
## 103 0.0072157919 2.929212e-04 2.252390e+00
## 104 0.0079357766 3.065992e-04 2.449175e+00
## 105 0.0043313083 1.884715e-04 1.431507e+00
## 106 0.0079463266 2.920331e-04 2.452038e+00
## 107 0.0073073390 2.673417e-04 2.277568e+00
## 108 0.0033127028 1.421974e-04 1.125659e+00
## 109 0.0065593510 2.360778e-04 2.070437e+00
## 110 0.0054729124 1.998401e-04 1.763375e+00
## 111 0.0052820408 1.890044e-04 1.708592e+00
## 112 0.0049510830 1.698197e-04 1.612957e+00
## 113 0.0057691261 1.870504e-04 1.847878e+00
## 114 0.0052404812 1.719513e-04 1.696629e+00
## 115 0.0031608312 1.135980e-04 1.079126e+00
## 116 0.0019218884 7.576162e-05 6.881045e-01
## 117 0.0051339892 1.531220e-04 1.665914e+00
## 118 0.0045205255 1.369571e-04 1.487245e+00
## 119 0.0049925933 1.445954e-04 1.624998e+00
## 120 0.0045310957 1.330491e-04 1.490350e+00
```

```
## 121 0.0045416996 1.287859e-04 1.493464e+00
## 122 0.0029835670 8.997247e-05 1.024470e+00
## 123 0.0049819093 1.312728e-04 1.621900e+00
## 124 0.0044415288 1.190159e-04 1.464013e+00
## 125 0.0038544567 1.032063e-04 1.289605e+00
## 126 0.0034025653 9.130474e-05 1.153069e+00
## 127 0.0008944010 2.948752e-05 3.419826e-01
## 128 0.0020597601 5.533352e-05 7.327813e-01
## 129 0.0006779126 2.120526e-05 2.649559e-01
## 130 0.0015534190 4.027889e-05 5.669159e-01
## 131 0.0012627577 3.228529e-05 4.691971e-01
## 132 0.0008613737 2.291500e-05 3.303545e-01
## 133 0.0028023471 5.684342e-05 9.681918e-01
## 134 0.0010340117 2.433609e-05 3.907046e-01
## 135 0.0007830997 1.887379e-05 3.026266e-01
## 136 0.0006496158 1.565414e-05 2.547346e-01
## 137 0.0004047949 1.013634e-05 1.644279e-01
## 138 0.0003939517 9.570122e-06 1.603363e-01
## 139 0.0003837916 9.092727e-06 1.564941e-01
## 140 0.0001351837 3.408385e-06 5.910950e-02
## 141 0.9988660793 2.273737e-05 1.202588e+00
## 142 0.9968733366 4.547474e-05 3.022515e+00
## 143 0.9449585650 1.364242e-04 3.542723e+01
## 144 0.9138436315 2.501110e-04 5.003100e+01
## 145 0.8742767886 3.637979e-04 6.578366e+01
## 146 0.9957467614 4.547474e-05 3.983609e+00
## 147 0.8515067793 3.637979e-04 7.371141e+01
## 148 0.8369657110 4.320100e-04 7.839592e+01
## 149 0.8269279431 3.865352e-04 8.146919e+01
## 150 0.8140151920 3.865352e-04 8.523901e+01
## 151 0.8113858837 3.865352e-04 8.598205e+01
## 152 0.9834007994 1.591616e-04 1.314757e+01
## 153 0.9883781740 1.818989e-04 9.668708e+00
## 154 0.9779119887 2.728484e-04 1.676152e+01
## 155 0.7870678940 4.774847e-04 9.247685e+01
## 156 0.7842971797 4.320100e-04 9.317487e+01
## 157 0.7730873165 4.320100e-04 9.591440e+01
## 158 0.9713851227 3.183231e-04 2.082275e+01
## 159 0.9706627714 3.410605e-04 2.125868e+01
## 160 0.7442545109 5.229595e-04 1.023592e+02
## 161 0.9781570054 3.410605e-04 1.660440e+01
## 162 0.7324376941 5.229595e-04 1.047599e+02
## 163 0.9920667139 1.364242e-04 6.925402e+00
## 164 0.9854742632 2.955858e-04 1.172491e+01
## 165 0.7204816995 5.229595e-04 1.070514e+02
## 166 0.9724820848 4.092726e-04 2.015588e+01
## 167 0.7114326149 5.456968e-04 1.086956e+02
## 168 0.7084019001 5.684342e-04 1.092292e+02
## 169 0.9840822645 3.183231e-04 1.268384e+01
## 170 0.9840241694 3.865352e-04 1.272352e+01
```

```
## 171 0.9669946928 5.684342e-04 2.343494e+01
## 172 0.9805932700 4.092726e-04 1.502161e+01
## 173 0.9847136050 3.865352e-04 1.225091e+01
## 174 0.9752714720 5.684342e-04 1.843241e+01
## 175 0.9776138813 5.911716e-04 1.695220e+01
## 176 0.9776664831 5.229595e-04 1.691859e+01
## 177 0.9652288293 7.958079e-04 2.446152e+01
## 178 0.9837164157 5.002221e-04 1.293325e+01
## 179 0.9718844660 7.958079e-04 2.051993e+01
## 180 0.9763075252 7.048584e-04 1.778158e+01
## 181 0.9593506991 1.023182e-03 2.778850e+01
## 182 0.9546031185 9.777068e-04 3.038274e+01
## 183 0.9577831725 1.045919e-03 2.865378e+01
## 184 0.9499815269 1.227818e-03 3.283579e+01
## 185 0.9466256517 1.386979e-03 3.457541e+01
## 186 0.9388502906 1.523404e-03 3.848176e+01
## 187 0.9394604558 1.455192e-03 3.818119e+01
## 188 0.9258376849 1.773515e-03 4.466981e+01
## 189 0.9186501586 1.887202e-03 4.791936e+01
## 190 0.9173773208 2.023626e-03 4.848322e+01
## 191 0.8971744546 2.228262e-03 5.700312e+01
## 192 0.9042537306 2.387424e-03 5.410614e+01
## 193 0.8967542453 2.501110e-03 5.717223e+01
## 194 0.8788605385 2.842171e-03 6.409419e+01
## 195 0.8670600016 3.069545e-03 6.837834e+01
## 196 0.8519199662 3.296918e-03 7.357414e+01
## 197 0.8314978174 3.569767e-03 8.008585e+01
## 198 0.8175605001 3.888090e-03 8.422408e+01
## 199 0.7990558220 4.138201e-03 8.935898e+01
## 200 0.7763903068 4.320100e-03 9.512116e+01
## 201 0.7540377453 4.524736e-03 1.002673e+02
## 202 0.7303958512 4.706635e-03 1.051609e+02
## 203 0.7054194498 5.047696e-03 1.097459e+02
## 204 0.6796368416 5.275069e-03 1.138748e+02
## 205 0.6526900177 5.388756e-03 1.175580e+02
## 206 0.6249575551 5.525180e-03 1.206950e+02
## 207 0.5966235892 5.661605e-03 1.232331e+02
## 208 0.5678155571 5.729817e-03 1.251368e+02
## 209 0.5388353361 5.820766e-03 1.263737e+02
## 210 0.5097887401 5.775291e-03 1.269375e+02
## 211 0.4809493265 5.809397e-03 1.268310e+02
## 212 0.4525860305 5.695711e-03 1.260779e+02
## 213 0.4246944081 5.616130e-03 1.247068e+02
## 214 0.3975029540 5.525180e-03 1.227616e+02
## 215 0.3714418534 5.377387e-03 1.203245e+02
## 216 0.3463726449 5.195488e-03 1.174405e+02
## 217 0.3222293100 4.968115e-03 1.141504e+02
## 218 0.2998378011 4.820322e-03 1.106369e+02
## 219 0.2782151802 4.627054e-03 1.068082e+02
## 220 0.2575336379 4.422418e-03 1.027313e+02
```

```
## 221 0.2385666238 4.240519e-03 9.862176e+01
## 222 0.2209536157 3.984724e-03 9.447485e+01
## 223 0.2047702122 3.774403e-03 9.037150e+01
## 224 0.1894896218 3.569767e-03 8.622765e+01
## 225 0.1751000602 3.370815e-03 8.207473e+01
## 226 0.1629109309 3.200284e-03 7.835735e+01
## 227 0.1513688343 3.018386e-03 7.466028e+01
## 228 0.1407920203 2.779643e-03 7.111383e+01
## 229 0.1311818909 2.637535e-03 6.775346e+01
## 230 0.1229080743 2.478373e-03 6.474995e+01
## 231 0.1145085940 2.319211e-03 6.159118e+01
## 232 0.1078176568 2.185629e-03 5.899216e+01
## 233 0.1013405252 2.054890e-03 5.640295e+01
## 234 0.0950506590 1.952571e-03 5.381650e+01
## 235 0.0900952108 1.813305e-03 5.172648e+01
## 236 0.0853053726 1.728040e-03 4.966060e+01
## 237 0.0806645751 1.628564e-03 4.761437e+01
## 238 0.0765529906 1.548983e-03 4.576325e+01
## 239 0.0728815697 1.458034e-03 4.407871e+01
## 240 0.0687681370 1.386979e-03 4.215450e+01
## 241 0.0657930997 1.313083e-03 4.073757e+01
## 242 0.0631533334 1.256240e-03 3.946192e+01
## 243 0.0600436741 1.173817e-03 3.793619e+01
## 244 0.0575729421 1.129763e-03 3.670558e+01
## 245 0.0551767124 1.077183e-03 3.549599e+01
## 246 0.0529277808 1.034550e-03 3.434586e+01
## 247 0.0512960727 9.819701e-04 3.350207e+01
## 248 0.0489329530 9.407586e-04 3.226572e+01
## 249 0.0473333382 8.967049e-04 3.141892e+01
## 250 0.0457041898 8.611778e-04 3.054800e+01
## 251 0.0438089568 8.256507e-04 2.952374e+01
## 252 0.0420355306 7.901235e-04 2.855415e+01
## 253 0.0410038643 7.588596e-04 2.798499e+01
## 254 0.0392323709 7.190692e-04 2.699865e+01
## 255 0.0378844982 6.906475e-04 2.624031e+01
## 256 0.0369537824 6.707523e-04 2.571259e+01
## 257 0.0358213189 6.465939e-04 2.506587e+01
## 258 0.0342592753 6.309619e-04 2.416529e+01
## 259 0.0334122262 6.053824e-04 2.367267e+01
## 260 0.0315984954 5.684342e-04 2.260747e+01
## 261 0.0319346255 5.542233e-04 2.280597e+01
## 262 0.0310767224 5.428547e-04 2.229833e+01
## 263 0.0300907280 5.215384e-04 2.171076e+01
## 264 0.0290509742 5.087486e-04 2.108623e+01
## 265 0.0281788315 4.867218e-04 2.055838e+01
## 266 0.0273246057 4.646949e-04 2.003774e+01
## 267 0.0254453608 4.412470e-04 1.887925e+01
## 268 0.0249332948 4.185097e-04 1.856033e+01
## 269 0.0242147792 4.085621e-04 1.811043e+01
## 270 0.0230714900 3.950618e-04 1.738856e+01
```

```
## 271 0.0223073609 3.765876e-04 1.690187e+01
## 272 0.0212523888 3.545608e-04 1.622416e+01
## 273 0.0207404313 3.446132e-04 1.589279e+01
## 274 0.0200941504 3.304024e-04 1.547209e+01
## 275 0.0193857669 3.197442e-04 1.500782e+01
## 276 0.0190468722 3.147704e-04 1.478452e+01
## 277 0.0181416212 2.948752e-04 1.418415e+01
## 278 0.0170981824 2.849276e-04 1.348485e+01
## 279 0.0170041999 2.813749e-04 1.342147e+01
## 280 0.0165549607 2.700062e-04 1.311758e+01
## 281 0.0161129893 2.557954e-04 1.281708e+01
## 282 0.0159626969 2.565059e-04 1.271455e+01
## 283 0.0151063110 2.422951e-04 1.212682e+01
## 284 0.0139995907 2.192024e-04 1.135817e+01
## 285 0.0136652227 2.131628e-04 1.112383e+01
## 286 0.0126242144 1.989520e-04 1.038760e+01
## 287 0.0123389053 1.961098e-04 1.018401e+01
## 288 0.0121735839 1.854517e-04 1.006567e+01
## 289 0.0114546179 1.769251e-04 9.547785e+00
## 290 0.0101588717 1.584510e-04 8.600356e+00
## 291 0.0101494856 1.563194e-04 8.593423e+00
## 292 0.0103482666 1.559641e-04 8.740023e+00
## 293 0.0094358480 1.442402e-04 8.063223e+00
## 294 0.0092241279 1.460165e-04 7.904712e+00
## 295 0.0092938940 1.442402e-04 7.957008e+00
## 296 0.0085282062 1.332268e-04 7.379578e+00
## 297 0.0077146532 1.193712e-04 6.757207e+00
## 298 0.0064733886 1.001865e-04 5.788148e+00
## 299 0.0067337695 1.030287e-04 5.993523e+00
## 300 0.0072323835 1.053380e-04 6.383638e+00
## 301 0.0063285005 9.467982e-05 5.673356e+00
## 302 0.0055950712 8.490986e-05 5.086357e+00
## 303 0.0055234715 8.277823e-05 5.028498e+00
## 304 0.0049757801 7.531753e-05 4.582400e+00
## 305 0.0031511083 4.973799e-05 3.043777e+00
## 306 0.0020674675 3.321787e-05 2.079655e+00
## 307 0.0032876233 4.760636e-05 3.162161e+00
## 308 0.0016093833 2.615685e-05 1.656053e+00
## 309 0.0015020194 2.384759e-05 1.555022e+00
## 310 0.0013233517 2.087219e-05 1.385201e+00
## 311 0.0012981466 2.029488e-05 1.361062e+00
## 312 0.0010873900 1.656453e-05 1.157283e+00
## 313 0.0009256793 1.429967e-05 9.983149e-01
## 314 0.0007653369 1.183498e-05 8.380612e-01
## 315 0.8245533718 3.183231e-04 1.125265e+02
## 316 0.7217819564 4.320100e-04 1.462539e+02
## 317 0.7206570365 3.410605e-04 1.465414e+02
## 318 0.9865791003 2.728484e-04 1.499657e+01
## 319 0.7071090499 4.092726e-04 1.498763e+02
## 320 0.7037094838 3.637979e-04 1.506764e+02
```

```
## 321 0.9791885714 3.637979e-04 2.182516e+01
## 322 0.9774029218 3.637979e-04 2.339707e+01
## 323 0.9489542697 7.503331e-04 4.569637e+01
## 324 0.9506353183 7.503331e-04 4.449272e+01
## 325 0.9608207347 7.275958e-04 3.692870e+01
## 326 0.9570742952 6.593837e-04 3.976768e+01
## 327 0.9527799727 8.412826e-04 4.293957e+01
## 328 0.9465948847 8.640200e-04 4.736596e+01
## 329 0.9398207749 1.045919e-03 5.203806e+01
## 330 0.9324104838 1.159606e-03 5.695731e+01
## 331 0.9236874519 1.296030e-03 6.251412e+01
## 332 0.9130207127 1.500666e-03 6.899632e+01
## 333 0.8984383646 1.614353e-03 7.735563e+01
## 334 0.8842991349 1.841727e-03 8.496065e+01
## 335 0.8647008622 2.000888e-03 9.476922e+01
## 336 0.8440863746 2.250999e-03 1.042550e+02
## 337 0.8252856435 2.569323e-03 1.122283e+02
## 338 0.8049005021 2.796696e-03 1.201979e+02
## 339 0.7982302361 2.933120e-03 1.226609e+02
## 340 0.7917132598 2.910383e-03 1.250007e+02
## 341 0.7825025500 2.955858e-03 1.281979e+02
## 342 0.7687531797 3.001333e-03 1.327376e+02
## 343 0.7561837288 3.092282e-03 1.366506e+02
## 344 0.7412117441 3.296918e-03 1.410255e+02
## 345 0.7197348862 3.296918e-03 1.467759e+02
## 346 0.7074772600 3.478817e-03 1.497888e+02
## 347 0.6717880501 3.592504e-03 1.574885e+02
## 348 0.6340104289 3.842615e-03 1.639644e+02
## 349 0.5946934857 3.910827e-03 1.689475e+02
## 350 0.5646289503 3.979039e-03 1.715822e+02
## 351 0.5545193528 3.979039e-03 1.722426e+02
## 352 0.5444010219 4.069989e-03 1.727908e+02
## 353 0.5342799966 4.069989e-03 1.732264e+02
## 354 0.5241626639 4.024514e-03 1.735497e+02
## 355 0.5140687688 4.024514e-03 1.737606e+02
## 356 0.5039638847 4.024514e-03 1.738601e+02
## 357 0.4939413782 4.024514e-03 1.738487e+02
## 358 0.4838763971 4.069989e-03 1.737267e+02
## 359 0.4739056067 4.001777e-03 1.734967e+02
## 360 0.4640228394 4.024514e-03 1.731612e+02
## 361 0.4540908327 3.967671e-03 1.727162e+02
## 362 0.4442001443 3.956302e-03 1.721652e+02
## 363 0.4344638268 3.967671e-03 1.715174e+02
## 364 0.4248662839 3.944933e-03 1.707761e+02
## 365 0.4153010678 3.899459e-03 1.699353e+02
## 366 0.4058283776 3.888090e-03 1.690018e+02
## 367 0.3964544220 3.853984e-03 1.679789e+02
## 368 0.3780307372 3.797140e-03 1.656781e+02
## 369 0.3691838327 3.740297e-03 1.644353e+02
## 370 0.3601841291 3.717560e-03 1.630783e+02
```

```
## 371 0.3513059113 3.694822e-03 1.616472e+02
## 372 0.3426680814 3.615241e-03 1.601659e+02
## 373 0.3341775539 3.615241e-03 1.586236e+02
## 374 0.3263307718 3.512923e-03 1.571216e+02
## 375 0.3176647908 3.524292e-03 1.553762e+02
## 376 0.3096579303 3.456080e-03 1.536820e+02
## 377 0.3016871278 3.410605e-03 1.519167e+02
## 378 0.2937445083 3.365130e-03 1.500786e+02
## 379 0.2645627893 3.149125e-03 1.426316e+02
## 380 0.2376658668 2.972911e-03 1.347637e+02
## 381 0.2133906031 2.717115e-03 1.267881e+02
## 382 0.1917275935 2.552270e-03 1.189262e+02
## 383 0.1719288812 2.336265e-03 1.110858e+02
## 384 0.1547283475 2.165734e-03 1.037313e+02
## 385 0.1397518005 1.983835e-03 9.688740e+01
## 386 0.1262766128 1.790568e-03 9.035427e+01
## 387 0.1148191852 1.656986e-03 8.449992e+01
## 388 0.1043104699 1.543299e-03 7.887112e+01
## 389 0.0952646944 1.389822e-03 7.381344e+01
## 390 0.0873933159 1.276135e-03 6.924141e+01
## 391 0.0803882206 1.173817e-03 6.502986e+01
## 392 0.0745150135 1.080025e-03 6.138865e+01
## 393 0.0688755389 1.011813e-03 5.779191e+01
## 394 0.0640291293 9.379164e-04 5.461765e+01
## 395 0.0599569282 8.782308e-04 5.188749e+01
## 396 0.0563437776 8.071765e-04 4.941434e+01
## 397 0.0527570138 7.602807e-04 4.690959e+01
## 398 0.0497139479 6.977530e-04 4.474378e+01
## 399 0.0480157292 6.721734e-04 4.351815e+01
## 400 0.0452491097 6.295409e-04 4.149431e+01
## 401 0.0426348264 5.854872e-04 3.954972e+01
## 402 0.0418889373 5.655920e-04 3.898895e+01
## 403 0.0382890192 5.314860e-04 3.624367e+01
## 404 0.0380780569 4.988010e-04 3.608073e+01
## 405 0.0359323956 4.732215e-04 3.441002e+01
## 406 0.0334360058 4.334311e-04 3.243414e+01
## 407 0.0321734505 4.106937e-04 3.142107e+01
## 408 0.0296632117 3.772982e-04 2.937791e+01
## 409 0.0262228570 3.247180e-04 2.651084e+01
## 410 0.0244717041 3.019807e-04 2.501946e+01
## 411 0.0225142902 2.792433e-04 2.332478e+01
## 412 0.0222902408 2.671641e-04 2.312885e+01
## 413 0.0171237543 1.953993e-04 1.848849e+01
## 414 0.0167565439 1.882938e-04 1.814896e+01
## 415 0.0136728066 1.669775e-04 1.523922e+01
## 416 0.0146676142 1.712408e-04 1.618985e+01
## 417 0.0144936912 1.648459e-04 1.602451e+01
## 418 0.0112687942 1.321609e-04 1.288935e+01
## 419 0.0115491230 1.278977e-04 1.316748e+01
## 420 0.0106241623 1.190159e-04 1.224535e+01
```

```
## 421 0.0104145957 1.158185e-04 1.203461e+01
## 422 0.0072279879 8.206769e-05 8.736471e+00
## 423 0.0075034410 8.260059e-05 9.029263e+00
## 424 0.0067661506 7.513989e-05 8.241832e+00
## 425 0.0061426662 6.785683e-05 7.566204e+00
## 426 0.0038915702 4.263256e-05 5.037691e+00
## 427 0.0014837738 1.660894e-05 2.105691e+00
## 428 0.9739705229 3.183231e-04 3.190811e+01
## 429 0.9581252846 4.774847e-04 4.720499e+01
## 430 0.9517360513 6.821210e-04 5.292099e+01
## 431 0.9314832982 9.549694e-04 6.970812e+01
## 432 0.9239407916 1.023182e-03 7.551698e+01
## 433 0.8829187254 1.500666e-03 1.037618e+02
## 434 0.8693495329 1.591616e-03 1.120417e+02
## 435 0.8540799888 1.705303e-03 1.208125e+02
## 436 0.8458546370 1.750777e-03 1.253117e+02
## 437 0.8182474812 2.046363e-03 1.393408e+02
## 438 0.7971419404 2.137313e-03 1.490284e+02
## 439 0.7694534406 2.319211e-03 1.604812e+02
## 440 0.7512312993 2.432898e-03 1.672858e+02
## 441 0.7257297705 2.614797e-03 1.758845e+02
## 442 0.6987743552 2.751221e-03 1.838535e+02
## 443 0.6687974076 2.796696e-03 1.914237e+02
## 444 0.6413862616 2.955858e-03 1.971992e+02
## 445 0.6116171232 3.024070e-03 2.022694e+02
## 446 0.5804275861 3.092282e-03 2.062718e+02
## 447 0.5565039703 3.069545e-03 2.084499e+02
## 448 0.5491803302 3.205969e-03 2.089634e+02
## 449 0.5408931872 3.115019e-03 2.094583e+02
## 450 0.5331477439 3.115019e-03 2.098382e+02
## 451 0.5253839777 3.092282e-03 2.101391e+02
## 452 0.5175198942 3.115019e-03 2.103624e+02
## 453 0.5098598911 3.115019e-03 2.105012e+02
## 454 0.5020316826 3.137757e-03 2.105627e+02
## 455 0.4943361915 3.149125e-03 2.105443e+02
## 456 0.4859435438 3.126388e-03 2.104348e+02
## 457 0.4778729328 3.183231e-03 2.102415e+02
## 458 0.4709640086 3.092282e-03 2.100075e+02
## 459 0.4632542448 3.115019e-03 2.096716e+02
## 460 0.4547327245 3.103651e-03 2.092085e+02
## 461 0.4481248999 3.080913e-03 2.087828e+02
## 462 0.4403449083 3.103651e-03 2.082069e+02
## 463 0.4244335395 3.103651e-03 2.067767e+02
## 464 0.4101296179 3.046807e-03 2.051998e+02
## 465 0.4031115020 3.024070e-03 2.043247e+02
## 466 0.3945570353 3.035439e-03 2.031671e+02
## 467 0.3884887099 3.001333e-03 2.022852e+02
## 468 0.3817066892 2.967226e-03 2.012395e+02
## 469 0.3665037121 2.967226e-03 1.986635e+02
## 470 0.3601003061 2.933120e-03 1.974818e+02
```

```
## 471 0.3468517224 2.876277e-03 1.948527e+02
## 472 0.3390351384 2.853540e-03 1.931841e+02
## 473 0.3132411524 2.739853e-03 1.870481e+02
## 474 0.2887339991 2.603429e-03 1.803003e+02
## 475 0.2656069006 2.489742e-03 1.730814e+02
## 476 0.2451003428 2.341949e-03 1.659612e+02
## 477 0.2246865651 2.256684e-03 1.581716e+02
## 478 0.2067354134 2.074785e-03 1.507160e+02
## 479 0.1892728810 1.995204e-03 1.428917e+02
## 480 0.1732825632 1.887202e-03 1.352060e+02
## 481 0.1601767883 1.762146e-03 1.285150e+02
## 482 0.1464249692 1.614353e-03 1.210932e+02
## 483 0.1351133727 1.512035e-03 1.146632e+02
## 484 0.1249110659 1.409717e-03 1.085974e+02
## 485 0.1149583842 1.327294e-03 1.024227e+02
## 486 0.1063133263 1.256240e-03 9.684105e+01
## 487 0.0990038062 1.151079e-03 9.195433e+01
## 488 0.0921324519 1.085709e-03 8.721300e+01
## 489 0.0856768408 1.014655e-03 8.262121e+01
## 490 0.0774871735 8.895995e-04 7.659259e+01
## 491 0.0712791503 8.384404e-04 7.186104e+01
## 492 0.0670826307 8.014922e-04 6.857851e+01
## 493 0.0632461733 7.588596e-04 6.551508e+01
## 494 0.0595179264 6.934897e-04 6.247810e+01
## 495 0.0566148355 6.494361e-04 6.007050e+01
## 496 0.0538034772 6.323830e-04 5.770173e+01
## 497 0.0486651783 5.670131e-04 5.327252e+01
## 498 0.0443861254 5.087486e-04 4.947898e+01
## 499 0.0423607413 4.774847e-04 4.764792e+01
## 500 0.0387558801 4.405365e-04 4.432894e+01
## 501 0.0362433622 3.979039e-04 4.196772e+01
## 502 0.0312978570 3.424816e-04 3.719511e+01
## 503 0.0291301558 3.161915e-04 3.504673e+01
## 504 0.0262344278 2.799538e-04 3.211807e+01
## 505 0.0222655160 2.344791e-04 2.798422e+01
## 506 0.0209212097 2.167155e-04 2.654956e+01
## 507 0.0187956187 1.854517e-04 2.424197e+01
## 508 0.0175053541 1.733724e-04 2.281619e+01
## 509 0.0170549584 1.740830e-04 2.231377e+01
## 510 0.0155318017 1.534772e-04 2.059554e+01
## 511 0.0122873467 1.175948e-04 1.682712e+01
## 512 0.0115115998 1.090683e-04 1.590160e+01
## 513 0.0063234299 5.702105e-05 9.401524e+00
## 514 0.0029974877 2.886580e-05 4.825386e+00
## 515 0.9879032950 1.818989e-04 1.905952e+01
## 516 0.9557986061 5.229595e-04 5.661723e+01
## 517 0.9479775406 6.593837e-04 6.450309e+01
## 518 0.8585554235 1.477929e-03 1.358219e+02
## 519 0.8430856155 1.546141e-03 1.455728e+02
## 520 0.7956366176 1.909939e-03 1.718587e+02
```

```
## 521 0.7906872463 1.978151e-03 1.743105e+02
## 522 0.7707994769 2.091838e-03 1.836488e+02
## 523 0.7656805599 2.114575e-03 1.859228e+02
## 524 0.7604374548 2.114575e-03 1.881983e+02
## 525 0.7497062722 2.160050e-03 1.926891e+02
## 526 0.6786102347 2.432898e-03 2.171006e+02
## 527 0.6535612658 2.432898e-03 2.236158e+02
## 528 0.6291762361 2.637535e-03 2.289678e+02
## 529 0.6161644751 2.683009e-03 2.314320e+02
## 530 0.6023955768 2.660272e-03 2.337471e+02
## 531 0.5752391394 2.683009e-03 2.374424e+02
## 532 0.5614785005 2.728484e-03 2.388783e+02
## 533 0.5477765035 2.728484e-03 2.400190e+02
## 534 0.5339967538 2.773959e-03 2.408763e+02
## 535 0.5271009957 2.705747e-03 2.411966e+02
## 536 0.5201969100 2.796696e-03 2.414448e+02
## 537 0.5136156176 2.705747e-03 2.416138e+02
## 538 0.5064146603 2.773959e-03 2.417234e+02
## 539 0.5000739492 2.694378e-03 2.417546e+02
## 540 0.4927013970 2.773959e-03 2.417142e+02
## 541 0.4865669280 2.705747e-03 2.416176e+02
## 542 0.4790542113 2.762590e-03 2.414214e+02
## 543 0.4730686516 2.671641e-03 2.412036e+02
## 544 0.4652896104 2.762590e-03 2.408390e+02
## 545 0.4596985707 2.660272e-03 2.405200e+02
## 546 0.4518731726 2.728484e-03 2.399934e+02
## 547 0.4464944061 2.705747e-03 2.395771e+02
## 548 0.4383882606 2.751221e-03 2.388658e+02
## 549 0.4332704697 2.717115e-03 2.383648e+02
## 550 0.4250602046 2.705747e-03 2.374767e+02
## 551 0.4121783070 2.660272e-03 2.358730e+02
## 552 0.3990886512 2.683009e-03 2.339788e+02
## 553 0.3862893556 2.648903e-03 2.318665e+02
## 554 0.3754237503 2.569323e-03 2.298698e+02
## 555 0.3616409930 2.535216e-03 2.270655e+02
## 556 0.3376148851 2.489742e-03 2.214398e+02
## 557 0.3148567191 2.387424e-03 2.152281e+02
## 558 0.2930650192 2.296474e-03 2.084520e+02
## 559 0.2725349583 2.239631e-03 2.013031e+02
## 560 0.2526609632 2.148681e-03 1.936500e+02
## 561 0.2340710094 2.052047e-03 1.858136e+02
## 562 0.2168796638 1.949729e-03 1.779593e+02
## 563 0.2009033818 1.864464e-03 1.701137e+02
## 564 0.1860491983 1.739409e-03 1.623251e+02
## 565 0.1721985913 1.631406e-03 1.546131e+02
## 566 0.1595489758 1.551825e-03 1.471722e+02
## 567 0.1483272987 1.472245e-03 1.402374e+02
## 568 0.1369489055 1.381295e-03 1.328689e+02
## 569 0.1277821889 1.307399e-03 1.266731e+02
## 570 0.1188200399 1.230660e-03 1.203801e+02
```

```
## 571 0.1108150607 1.159606e-03 1.145522e+02
## 572 0.1029735904 1.085709e-03 1.086442e+02
## 573 0.0969130691 1.008971e-03 1.039362e+02
## 574 0.0910159397 9.549694e-04 9.923045e+01
## 575 0.0809054998 8.526513e-04 9.085943e+01
## 576 0.0760408065 8.242296e-04 8.668666e+01
## 577 0.0721450360 7.730705e-04 8.327291e+01
## 578 0.0646635868 6.878054e-04 7.652689e+01
## 579 0.0584089594 6.124878e-04 7.068169e+01
## 580 0.0425665291 4.334311e-04 5.492041e+01
## 581 0.0368242790 3.723244e-04 4.881500e+01
## 582 0.0327653161 3.254286e-04 4.435120e+01
## 583 0.0303672556 3.019807e-04 4.165066e+01
## 584 0.0255317764 2.408740e-04 3.604722e+01
## 585 0.0220928084 2.074785e-04 3.191883e+01
## 586 0.0190242438 1.747935e-04 2.812044e+01
## 587 0.0182453835 1.641354e-04 2.713739e+01
## 588 0.0146786794 1.342926e-04 2.252568e+01
## 589 0.0148342673 1.318057e-04 2.273090e+01
## 590 0.0067747384 5.595524e-05 1.147265e+01
## 591 0.0039729277 3.552714e-05 7.135662e+00
## 592 0.9376120170 1.364242e-04 8.291975e+01
## 593 0.8131323898 2.728484e-04 1.813504e+02
## 594 0.7842852796 2.614797e-04 1.976601e+02
## 595 0.7725995721 2.160050e-04 2.037116e+02
## 596 0.9899512736 1.136868e-04 1.807140e+01
## 597 0.9892410208 1.364242e-04 1.917988e+01
## 598 0.9905974548 4.547474e-05 1.705187e+01
## 599 0.9899607187 1.136868e-04 1.805657e+01
## 600 0.9881978428 2.046363e-04 2.078638e+01
## 601 0.9871230975 1.136868e-04 2.241653e+01
## 602 0.9858671672 1.136868e-04 2.429183e+01
## 603 0.9830353554 2.501110e-04 2.841461e+01
## 604 0.9813373651 2.046363e-04 3.082361e+01
## 605 0.9773468548 2.955858e-04 3.632239e+01
## 606 0.9739069333 2.728484e-04 4.090037e+01
## 607 0.9718032277 2.955858e-04 4.363403e+01
## 608 0.9717054786 2.728484e-04 4.375991e+01
## 609 0.9693244181 3.410605e-04 4.679595e+01
## 610 0.9674056469 3.637979e-04 4.920215e+01
## 611 0.9657804219 4.320100e-04 5.121348e+01
## 612 0.9640294581 4.320100e-04 5.335413e+01
## 613 0.9615903127 4.092726e-04 5.629267e+01
## 614 0.9557401063 5.002221e-04 6.314862e+01
## 615 0.9526019869 5.684342e-04 6.672256e+01
## 616 0.9555379813 5.684342e-04 6.338091e+01
## 617 0.9498887914 5.229595e-04 6.975818e+01
## 618 0.9477379922 6.366463e-04 7.213027e+01
## 619 0.9440873271 6.139089e-04 7.609010e+01
## 620 0.9413009855 6.593837e-04 7.905844e+01
```

```
## 621 0.9347171112 6.821210e-04 8.589731e+01
## 622 0.9356410078 7.275958e-04 8.495194e+01
## 623 0.9241238058 8.185452e-04 9.642460e+01
## 624 0.9167109317 9.322321e-04 1.034718e+02
## 625 0.9002775971 9.777068e-04 1.182519e+02
## 626 0.8975105254 1.023182e-03 1.206345e+02
## 627 0.8948410220 1.023182e-03 1.229056e+02
## 628 0.8849514302 1.091394e-03 1.310927e+02
## 629 0.8781973452 1.182343e-03 1.364872e+02
## 630 0.8713739066 1.318767e-03 1.417823e+02
## 631 0.8653106863 1.205080e-03 1.463617e+02
## 632 0.8568551243 1.318767e-03 1.525574e+02
## 633 0.8439977307 1.432454e-03 1.615717e+02
## 634 0.8382882243 1.455192e-03 1.654235e+02
## 635 0.8322022848 1.523404e-03 1.694304e+02
## 636 0.8223970372 1.591616e-03 1.756773e+02
## 637 0.8077108134 1.659828e-03 1.845697e+02
## 638 0.7963921928 1.750777e-03 1.910580e+02
## 639 0.7920823612 1.841727e-03 1.934477e+02
## 640 0.7723870044 1.773515e-03 2.038188e+02
## 641 0.7621278064 1.909939e-03 2.088745e+02
## 642 0.7533866233 1.955414e-03 2.130006e+02
## 643 0.7429182381 1.932676e-03 2.177271e+02
## 644 0.7324630781 1.978151e-03 2.222186e+02
## 645 0.7116572942 2.046363e-03 2.304936e+02
## 646 0.7015972710 2.182787e-03 2.341857e+02
## 647 0.6905548787 2.205525e-03 2.380117e+02
## 648 0.6785749952 2.228262e-03 2.418984e+02
## 649 0.6670714970 2.228262e-03 2.453760e+02
## 650 0.6557148413 2.273737e-03 2.485679e+02
## 651 0.6439860908 2.319211e-03 2.516162e+02
## 652 0.6322871661 2.296474e-03 2.544084e+02
## 653 0.6202949612 2.319211e-03 2.570161e+02
## 654 0.5962375826 2.364686e-03 2.614804e+02
## 655 0.5841022073 2.432898e-03 2.633485e+02
## 656 0.5718205398 2.410161e-03 2.649795e+02
## 657 0.5595743892 2.455636e-03 2.663474e+02
## 658 0.5472702048 2.478373e-03 2.674632e+02
## 659 0.5349480989 2.455636e-03 2.683221e+02
## 660 0.5226099893 2.455636e-03 2.689236e+02
## 661 0.5102124953 2.432898e-03 2.692680e+02
## 662 0.4979422954 2.478373e-03 2.693527e+02
## 663 0.4856726997 2.489742e-03 2.691825e+02
## 664 0.4734034059 2.489742e-03 2.687575e+02
## 665 0.4612181726 2.455636e-03 2.680825e+02
## 666 0.4488990902 2.489742e-03 2.671435e+02
## 667 0.4369826155 2.489742e-03 2.659888e+02
## 668 0.4252209050 2.421530e-03 2.646104e+02
## 669 0.4134060032 2.387424e-03 2.629858e+02
## 670 0.4016392325 2.387424e-03 2.611273e+02
```

```
## 671 0.3785769451 2.376055e-03 2.567817e+02
## 672 0.3561450234 2.285105e-03 2.516489e+02
## 673 0.3349222817 2.216893e-03 2.459536e+02
## 674 0.3141492203 2.194156e-03 2.395704e+02
## 675 0.2941696295 2.103206e-03 2.326563e+02
## 676 0.2756692140 2.000888e-03 2.255569e+02
## 677 0.2576535786 1.966782e-03 2.179787e+02
## 678 0.2404366600 1.892886e-03 2.101021e+02
## 679 0.2241807601 1.784883e-03 2.020751e+02
## 680 0.2082815345 1.733724e-03 1.936475e+02
## 681 0.1945229303 1.637090e-03 1.858749e+02
## 682 0.1812263363 1.563194e-03 1.779222e+02
## 683 0.1689728637 1.494982e-03 1.701926e+02
## 684 0.1576526827 1.404032e-03 1.626945e+02
## 685 0.1468041643 1.330136e-03 1.551720e+02
## 686 0.1367522796 1.267608e-03 1.478935e+02
## 687 0.1282099694 1.199396e-03 1.414637e+02
## 688 0.1196497944 1.128342e-03 1.347845e+02
## 689 0.1122979825 1.068656e-03 1.288507e+02
## 690 0.1054558270 1.014655e-03 1.231565e+02
## 691 0.0987411312 9.720225e-04 1.173999e+02
## 692 0.0925169226 9.151790e-04 1.119076e+02
## 693 0.0873658780 8.412826e-04 1.072432e+02
## 694 0.0827192174 7.929657e-04 1.029389e+02
## 695 0.0781781895 7.787548e-04 9.863999e+01
## 696 0.0741315493 7.190692e-04 9.472894e+01
## 697 0.0703762808 6.934897e-04 9.102905e+01
## 698 0.0669060951 6.508571e-04 8.754730e+01
## 699 0.0637169440 6.281198e-04 8.429228e+01
## 700 0.0605994572 5.883294e-04 8.105723e+01
## 701 0.0396045316 3.780087e-04 5.771440e+01
## 702 0.0292358452 2.671641e-04 4.496700e+01
## 703 0.0269540140 2.394529e-04 4.202491e+01
## 704 0.0250153862 2.302158e-04 3.948107e+01
## 705 0.0219899147 1.904255e-04 3.542320e+01
## 706 0.0198356418 1.683986e-04 3.246267e+01
## 707 0.0187112426 1.584510e-04 3.089192e+01
## 708 0.0174605653 1.513456e-04 2.912280e+01
## 709 0.0161187392 1.399769e-04 2.719745e+01
## 710 0.0142578099 1.190159e-04 2.447678e+01
## 711 0.0118405431 1.016076e-04 2.084503e+01
## 712 0.0100190372 8.135714e-05 1.802479e+01
## 713 0.0087020762 6.856737e-05 1.593405e+01
## 714 0.0080232943 6.394885e-05 1.483745e+01
## 715 0.0066613745 5.293543e-05 1.259330e+01
```

```
theo
```

```
##     Delta        Gamma         Vega
## 1       1 2.965737e-15 7.510937e-11
```

```
## 2       1 6.405520e-09 1.622243e-04
## 3       1 6.345760e-05 1.607108e+00
## 4       1 9.099099e-05 2.304410e+00
## 5       1 1.536594e-04 3.891532e+00
## 6       1 1.633432e-04 4.136780e+00
## 7       1 4.589754e-04 1.162387e+01
## 8       1 5.419483e-04 1.372522e+01
## 9       1 5.749236e-04 1.456034e+01
## 10      1 5.861281e-04 1.484410e+01
## 11      1 6.319591e-04 1.600481e+01
## 12      1 6.793095e-04 1.720399e+01
## 13      1 7.035082e-04 1.781684e+01
## 14      1 7.280291e-04 1.843785e+01
## 15      1 8.288896e-04 2.099221e+01
## 16      1 9.067666e-04 2.296450e+01
## 17      1 9.198746e-04 2.329647e+01
## 18      1 9.593484e-04 2.429617e+01
## 19      1 9.857530e-04 2.496488e+01
## 20      1 9.989707e-04 2.529963e+01
## 21      1 1.038641e-03 2.630430e+01
## 22      1 1.091429e-03 2.764121e+01
## 23      1 1.143869e-03 2.896928e+01
## 24      1 1.169878e-03 2.962799e+01
## 25      1 1.195705e-03 3.028207e+01
## 26      1 1.246683e-03 3.157312e+01
## 27      1 5.366710e-04 4.990574e+00
## 28      1 1.296547e-03 3.283596e+01
## 29      1 1.320982e-03 3.345480e+01
## 30      1 1.345045e-03 3.406421e+01
## 31      1 1.368705e-03 3.466341e+01
## 32      1 1.391931e-03 3.525163e+01
## 33      1 1.414694e-03 3.582812e+01
## 34      1 1.436965e-03 3.639214e+01
## 35      1 1.458716e-03 3.694299e+01
## 36      1 1.479918e-03 3.747995e+01
## 37      1 1.500545e-03 3.800235e+01
## 38      1 1.510635e-03 3.825789e+01
## 39      1 1.520571e-03 3.850953e+01
## 40      1 1.539972e-03 3.900085e+01
## 41      1 1.549429e-03 3.924038e+01
## 42      1 1.558721e-03 3.947571e+01
## 43      1 1.567845e-03 3.970678e+01
## 44      1 1.576798e-03 3.993351e+01
## 45      1 1.585577e-03 4.015584e+01
## 46      1 1.594179e-03 4.037370e+01
## 47      1 1.602602e-03 4.058703e+01
## 48      1 1.610844e-03 4.079575e+01
## 49      1 1.618902e-03 4.099981e+01
## 50      1 1.626773e-03 4.119915e+01
## 51      1 1.634455e-03 4.139371e+01
```

```
## 52       1 1.641946e-03 4.158344e+01
## 53       1 1.649245e-03 4.176828e+01
## 54       1 1.656348e-03 4.194817e+01
## 55       1 1.669961e-03 4.229293e+01
## 56       1 1.682771e-03 4.261734e+01
## 57       1 1.694764e-03 4.292107e+01
## 58       1 1.705927e-03 4.320379e+01
## 59       1 1.716250e-03 4.346524e+01
## 60       1 1.725724e-03 4.370515e+01
## 61       1 2.707976e-03 5.177227e+00
## 62       1 6.335886e-03 1.423586e+01
## 63       1 9.297366e-03 2.228686e+01
## 64       1 1.188165e-02 2.986030e+01
## 65       1 1.419205e-02 3.570985e+01
## 66       1 1.579997e-02 4.073182e+01
## 67       1 1.637239e-02 4.387623e+01
## 68       1 1.685028e-02 4.489239e+01
## 69       1 1.571991e-02 4.403476e+01
## 70       1 1.445636e-02 4.168714e+01
## 71       1 1.286602e-02 3.840163e+01
## 72       1 1.122238e-02 3.460208e+01
## 73       1 9.631209e-03 3.064451e+01
## 74       1 8.165151e-03 2.694289e+01
## 75       1 6.891668e-03 2.369007e+01
## 76       1 5.791126e-03 2.057422e+01
## 77       1 4.889370e-03 1.808291e+01
## 78       1 4.112969e-03 1.571531e+01
## 79       1 3.518728e-03 1.403546e+01
## 80       1 2.927171e-03 1.191277e+01
## 81       1 2.471423e-03 1.034959e+01
## 82       1 2.094529e-03 9.023429e+00
## 83       1 1.868725e-03 8.440607e+00
## 84       1 1.701887e-03 8.084887e+00
## 85       1 1.453887e-03 7.055746e+00
## 86       1 1.206474e-03 5.918775e+00
## 87       1 1.181719e-03 6.174959e+00
## 88       1 1.014835e-03 5.395675e+00
## 89       1 8.879334e-04 4.826026e+00
## 90       1 8.224522e-04 4.635279e+00
## 91       1 7.214005e-04 4.145072e+00
## 92       1 6.037469e-04 3.493044e+00
## 93       1 6.109443e-04 3.723391e+00
## 94       1 5.631561e-04 3.526908e+00
## 95       1 4.995112e-04 3.181201e+00
## 96       1 4.033044e-04 2.558043e+00
## 97       1 4.669057e-04 3.186383e+00
## 98       1 3.897048e-04 2.659354e+00
## 99       1 4.279919e-04 3.097839e+00
## 100      1 2.606327e-04 1.770578e+00
## 101      1 3.317990e-04 2.445427e+00
```

```
## 102      1 2.231321e-04 1.579102e+00
## 103      1 2.928826e-04 2.252390e+00
## 104      1 3.057063e-04 2.449174e+00
## 105      1 1.892576e-04 1.431507e+00
## 106      1 2.906359e-04 2.452037e+00
## 107      1 2.666419e-04 2.277568e+00
## 108      1 1.428524e-04 1.125659e+00
## 109      1 2.347628e-04 2.070437e+00
## 110      1 2.003894e-04 1.763375e+00
## 111      1 1.866880e-04 1.708592e+00
## 112      1 1.704518e-04 1.612957e+00
## 113      1 1.874657e-04 1.847878e+00
## 114      1 1.709504e-04 1.696628e+00
## 115      1 1.136593e-04 1.079126e+00
## 116      1 7.522131e-05 6.881044e-01
## 117      1 1.533674e-04 1.665914e+00
## 118      1 1.368035e-04 1.487245e+00
## 119      1 1.450444e-04 1.624998e+00
## 120      1 1.324713e-04 1.490350e+00
## 121      1 1.284391e-04 1.493463e+00
## 122      1 8.989167e-05 1.024470e+00
## 123      1 1.315076e-04 1.621900e+00
## 124      1 1.186848e-04 1.464013e+00
## 125      1 1.033462e-04 1.289605e+00
## 126      1 9.119248e-05 1.153069e+00
## 127      1 2.949359e-05 3.419826e-01
## 128      1 5.524069e-05 7.327813e-01
## 129      1 2.121697e-05 2.649559e-01
## 130      1 4.008834e-05 5.669159e-01
## 131      1 3.244268e-05 4.691971e-01
## 132      1 2.273452e-05 3.303545e-01
## 133      1 5.678058e-05 9.681918e-01
## 134      1 2.448691e-05 3.907046e-01
## 135      1 1.877719e-05 3.026265e-01
## 136      1 1.552931e-05 2.547346e-01
## 137      1 1.009395e-05 1.644279e-01
## 138      1 9.557914e-06 1.603363e-01
## 139      1 9.068957e-06 1.564941e-01
## 140      1 3.401845e-06 5.910949e-02
## 141      1 5.935613e-06 1.202588e+00
## 142      1 1.491822e-05 3.022515e+00
## 143      1 1.748581e-04 3.542723e+01
## 144      1 2.469379e-04 5.003100e+01
## 145      1 3.246883e-04 6.578366e+01
## 146      1 5.538831e-05 3.983609e+00
## 147      1 3.638173e-04 7.371141e+01
## 148      1 3.869386e-04 7.839592e+01
## 149      1 4.021073e-04 8.146919e+01
## 150      1 4.207140e-04 8.523901e+01
## 151      1 4.243814e-04 8.598205e+01
```

```
## 152      1 1.885291e-04 1.314757e+01
## 153      1 1.523925e-04 9.668708e+00
## 154      1 2.397086e-04 1.676152e+01
## 155      1 4.564378e-04 9.247685e+01
## 156      1 4.598830e-04 9.317487e+01
## 157      1 4.734045e-04 9.591440e+01
## 158      1 3.502821e-04 2.082275e+01
## 159      1 3.681818e-04 2.125868e+01
## 160      1 5.052142e-04 1.023592e+02
## 161      1 3.310574e-04 1.660440e+01
## 162      1 5.170631e-04 1.047599e+02
## 163      1 1.758912e-04 6.925401e+00
## 164      1 2.741664e-04 1.172491e+01
## 165      1 5.283732e-04 1.070514e+02
## 166      1 4.483709e-04 2.015588e+01
## 167      1 5.364887e-04 1.086956e+02
## 168      1 5.391223e-04 1.092292e+02
## 169      1 3.464448e-04 1.268384e+01
## 170      1 3.641706e-04 1.272352e+01
## 171      1 5.865659e-04 2.343494e+01
## 172      1 4.347695e-04 1.502161e+01
## 173      1 3.811569e-04 1.225091e+01
## 174      1 5.335701e-04 1.843241e+01
## 175      1 5.295203e-04 1.695219e+01
## 176      1 5.438416e-04 1.691859e+01
## 177      1 7.297482e-04 2.446152e+01
## 178      1 4.843664e-04 1.293325e+01
## 179      1 7.298274e-04 2.051993e+01
## 180      1 6.795286e-04 1.778157e+01
## 181      1 9.634068e-04 2.778850e+01
## 182      1 1.058003e-03 3.038274e+01
## 183      1 1.056522e-03 2.865378e+01
## 184      1 1.245126e-03 3.283579e+01
## 185      1 1.338678e-03 3.457541e+01
## 186      1 1.487502e-03 3.848176e+01
## 187      1 1.547606e-03 3.818119e+01
## 188      1 1.764618e-03 4.466981e+01
## 189      1 1.916598e-03 4.791936e+01
## 190      1 2.026745e-03 4.848322e+01
## 191      1 2.285620e-03 5.700312e+01
## 192      1 2.369323e-03 5.410614e+01
## 193      1 2.565597e-03 5.717224e+01
## 194      1 2.827204e-03 6.409419e+01
## 195      1 3.061859e-03 6.837834e+01
## 196      1 3.313655e-03 7.357414e+01
## 197      1 3.566138e-03 8.008585e+01
## 198      1 3.833729e-03 8.422408e+01
## 199      1 4.100876e-03 8.935898e+01
## 200      1 4.345394e-03 9.512116e+01
## 201      1 4.591201e-03 1.002673e+02
```

```
## 202      1 4.824350e-03 1.051609e+02
## 203      1 5.038064e-03 1.097459e+02
## 204      1 5.237291e-03 1.138748e+02
## 205      1 5.407359e-03 1.175580e+02
## 206      1 5.549803e-03 1.206950e+02
## 207      1 5.662838e-03 1.232331e+02
## 208      1 5.739996e-03 1.251368e+02
## 209      1 5.790830e-03 1.263737e+02
## 210      1 5.783436e-03 1.269375e+02
## 211      1 5.770808e-03 1.268310e+02
## 212      1 5.706618e-03 1.260779e+02
## 213      1 5.623239e-03 1.247068e+02
## 214      1 5.511802e-03 1.227616e+02
## 215      1 5.365399e-03 1.203245e+02
## 216      1 5.200749e-03 1.174405e+02
## 217      1 5.023780e-03 1.141504e+02
## 218      1 4.822156e-03 1.106369e+02
## 219      1 4.620147e-03 1.068082e+02
## 220      1 4.414589e-03 1.027313e+02
## 221      1 4.199671e-03 9.862176e+01
## 222      1 3.983699e-03 9.447486e+01
## 223      1 3.769109e-03 9.037150e+01
## 224      1 3.560205e-03 8.622766e+01
## 225      1 3.357108e-03 8.207474e+01
## 226      1 3.160186e-03 7.835735e+01
## 227      1 2.972663e-03 7.466029e+01
## 228      1 2.794586e-03 7.111383e+01
## 229      1 2.626476e-03 6.775347e+01
## 230      1 2.470345e-03 6.474995e+01
## 231      1 2.320621e-03 6.159118e+01
## 232      1 2.185434e-03 5.899216e+01
## 233      1 2.057736e-03 5.640296e+01
## 234      1 1.936536e-03 5.381650e+01
## 235      1 1.829165e-03 5.172648e+01
## 236      1 1.727906e-03 4.966060e+01
## 237      1 1.632071e-03 4.761437e+01
## 238      1 1.544238e-03 4.576325e+01
## 239      1 1.463614e-03 4.407871e+01
## 240      1 1.382474e-03 4.215450e+01
## 241      1 1.314101e-03 4.073757e+01
## 242      1 1.251589e-03 3.946192e+01
## 243      1 1.187365e-03 3.793619e+01
## 244      1 1.131323e-03 3.670558e+01
## 245      1 1.078125e-03 3.549600e+01
## 246      1 1.028283e-03 3.434586e+01
## 247      1 9.862443e-04 3.350207e+01
## 248      1 9.387208e-04 3.226572e+01
## 249      1 9.002285e-04 3.141892e+01
## 250      1 8.628615e-04 3.054800e+01
## 251      1 8.241004e-04 2.952374e+01
```

```
## 252      1 7.877129e-04 2.855415e+01
## 253      1 7.599986e-04 2.798499e+01
## 254      1 7.257160e-04 2.699865e+01
## 255      1 6.967127e-04 2.624031e+01
## 256      1 6.729124e-04 2.571259e+01
## 257      1 6.478320e-04 2.506587e+01
## 258      1 6.190076e-04 2.416529e+01
## 259      1 5.983618e-04 2.367267e+01
## 260      1 5.681393e-04 2.260747e+01
## 261      1 5.612390e-04 2.280597e+01
## 262      1 5.423048e-04 2.229833e+01
## 263      1 5.225657e-04 2.171076e+01
## 264      1 5.027534e-04 2.108623e+01
## 265      1 4.851582e-04 2.055838e+01
## 266      1 4.681835e-04 2.003774e+01
## 267      1 4.336040e-04 1.887924e+01
## 268      1 4.213968e-04 1.856033e+01
## 269      1 4.073321e-04 1.811043e+01
## 270      1 3.890420e-04 1.738856e+01
## 271      1 3.750852e-04 1.690187e+01
## 272      1 3.530803e-04 1.622416e+01
## 273      1 3.426208e-04 1.589279e+01
## 274      1 3.309513e-04 1.547209e+01
## 275      1 3.188278e-04 1.500782e+01
## 276      1 3.108954e-04 1.478452e+01
## 277      1 2.970389e-04 1.418415e+01
## 278      1 2.817919e-04 1.348485e+01
## 279      1 2.771340e-04 1.342147e+01
## 280      1 2.687356e-04 1.311758e+01
## 281      1 2.605723e-04 1.281708e+01
## 282      1 2.557337e-04 1.271455e+01
## 283      1 2.433333e-04 1.212682e+01
## 284      1 2.228733e-04 1.135817e+01
## 285      1 2.166665e-04 1.112382e+01
## 286      1 2.027348e-04 1.038760e+01
## 287      1 1.951111e-04 1.018401e+01
## 288      1 1.871085e-04 1.006567e+01
## 289      1 1.773809e-04 9.547784e+00
## 290      1 1.595337e-04 8.600355e+00
## 291      1 1.578182e-04 8.593422e+00
## 292      1 1.584262e-04 8.740022e+00
## 293      1 1.468081e-04 8.063223e+00
## 294      1 1.430155e-04 7.904712e+00
## 295      1 1.424121e-04 7.957007e+00
## 296      1 1.325286e-04 7.379577e+00
## 297      1 1.197495e-04 6.757206e+00
## 298      1 1.005628e-04 5.788147e+00
## 299      1 1.017943e-04 5.993523e+00
## 300      1 1.064282e-04 6.383637e+00
## 301      1 9.477734e-05 5.673356e+00
```

```
## 302      1 8.569488e-05 5.086357e+00
## 303      1 8.350427e-05 5.028497e+00
## 304      1 7.476262e-05 4.582399e+00
## 305      1 4.950430e-05 3.043777e+00
## 306      1 3.355059e-05 2.079655e+00
## 307      1 4.717848e-05 3.162160e+00
## 308      1 2.604597e-05 1.656053e+00
## 309      1 2.403122e-05 1.555021e+00
## 310      1 2.116811e-05 1.385200e+00
## 311      1 2.036345e-05 1.361061e+00
## 312      1 1.650002e-05 1.157283e+00
## 313      1 1.415489e-05 9.983146e-01
## 314      1 1.184942e-05 8.380610e-01
## 315      1 2.962116e-04 1.125265e+02
## 316      1 3.849948e-04 1.462539e+02
## 317      1 3.857516e-04 1.465414e+02
## 318      1 2.492562e-04 1.499657e+01
## 319      1 3.945303e-04 1.498763e+02
## 320      1 3.966365e-04 1.506764e+02
## 321      1 3.763482e-04 2.182516e+01
## 322      1 4.051306e-04 2.339707e+01
## 323      1 7.195256e-04 4.569637e+01
## 324      1 7.439813e-04 4.449272e+01
## 325      1 6.770998e-04 3.692870e+01
## 326      1 7.298201e-04 3.976768e+01
## 327      1 8.094228e-04 4.293957e+01
## 328      1 9.104989e-04 4.736596e+01
## 329      1 1.021977e-03 5.203806e+01
## 330      1 1.145156e-03 5.695731e+01
## 331      1 1.285580e-03 6.251412e+01
## 332      1 1.446950e-03 6.899632e+01
## 333      1 1.636487e-03 7.735564e+01
## 334      1 1.832429e-03 8.496066e+01
## 335      1 2.052238e-03 9.476922e+01
## 336      1 2.278795e-03 1.042550e+02
## 337      1 2.521143e-03 1.122283e+02
## 338      1 2.787624e-03 1.201979e+02
## 339      1 2.853015e-03 1.226609e+02
## 340      1 2.920148e-03 1.250007e+02
## 341      1 2.973185e-03 1.281979e+02
## 342      1 2.992392e-03 1.327376e+02
## 343      1 3.129191e-03 1.366506e+02
## 344      1 3.253189e-03 1.410255e+02
## 345      1 3.303230e-03 1.467759e+02
## 346      1 3.470016e-03 1.497888e+02
## 347      1 3.666233e-03 1.574885e+02
## 348      1 3.826898e-03 1.639644e+02
## 349      1 3.944270e-03 1.689475e+02
## 350      1 4.004695e-03 1.715822e+02
## 351      1 4.017222e-03 1.722426e+02
```

```
## 352      1 4.027182e-03 1.727908e+02
## 353      1 4.034544e-03 1.732264e+02
## 354      1 4.039329e-03 1.735497e+02
## 355      1 4.028730e-03 1.737606e+02
## 356      1 4.041187e-03 1.738601e+02
## 357      1 4.030602e-03 1.738487e+02
## 358      1 4.030263e-03 1.737267e+02
## 359      1 4.019740e-03 1.734967e+02
## 360      1 4.004255e-03 1.731612e+02
## 361      1 3.993830e-03 1.727162e+02
## 362      1 3.980922e-03 1.721652e+02
## 363      1 3.960651e-03 1.715174e+02
## 364      1 3.935630e-03 1.707761e+02
## 365      1 3.910785e-03 1.699353e+02
## 366      1 3.883702e-03 1.690018e+02
## 367      1 3.854449e-03 1.679789e+02
## 368      1 3.789637e-03 1.656781e+02
## 369      1 3.749899e-03 1.644353e+02
## 370      1 3.714779e-03 1.630783e+02
## 371      1 3.677717e-03 1.616472e+02
## 372      1 3.636792e-03 1.601659e+02
## 373      1 3.594189e-03 1.586237e+02
## 374      1 3.542503e-03 1.571216e+02
## 375      1 3.504353e-03 1.553762e+02
## 376      1 3.457300e-03 1.536820e+02
## 377      1 3.410665e-03 1.519167e+02
## 378      1 3.364350e-03 1.500786e+02
## 379      1 3.160245e-03 1.426316e+02
## 380      1 2.951787e-03 1.347638e+02
## 381      1 2.741535e-03 1.267881e+02
## 382      1 2.534645e-03 1.189262e+02
## 383      1 2.335838e-03 1.110858e+02
## 384      1 2.146622e-03 1.037313e+02
## 385      1 1.969962e-03 9.688741e+01
## 386      1 1.805991e-03 9.035427e+01
## 387      1 1.656662e-03 8.449992e+01
## 388      1 1.518968e-03 7.887113e+01
## 389      1 1.394690e-03 7.381344e+01
## 390      1 1.282742e-03 6.924142e+01
## 391      1 1.181381e-03 6.502986e+01
## 392      1 1.091912e-03 6.138865e+01
## 393      1 1.008665e-03 5.779192e+01
## 394      1 9.346359e-04 5.461766e+01
## 395      1 8.695497e-04 5.188749e+01
## 396      1 8.110320e-04 4.941434e+01
## 397      1 7.556779e-04 4.690960e+01
## 398      1 7.068469e-04 4.474378e+01
## 399      1 6.702645e-04 4.351815e+01
## 400      1 6.279991e-04 4.149431e+01
## 401      1 5.886755e-04 3.954972e+01
```

```
## 402      1 5.650378e-04 3.898895e+01
## 403      1 5.214116e-04 3.624367e+01
## 404      1 5.048290e-04 3.608073e+01
## 405      1 4.747664e-04 3.441002e+01
## 406      1 4.314161e-04 3.243414e+01
## 407      1 4.110185e-04 3.142106e+01
## 408      1 3.726915e-04 2.937790e+01
## 409      1 3.225556e-04 2.651084e+01
## 410      1 2.959141e-04 2.501946e+01
## 411      1 2.748775e-04 2.332478e+01
## 412      1 2.676806e-04 2.312885e+01
## 413      1 1.979195e-04 1.848849e+01
## 414      1 1.918070e-04 1.814896e+01
## 415      1 1.641719e-04 1.523922e+01
## 416      1 1.695738e-04 1.618985e+01
## 417      1 1.655555e-04 1.602451e+01
## 418      1 1.326617e-04 1.288935e+01
## 419      1 1.275286e-04 1.316748e+01
## 420      1 1.185560e-04 1.224535e+01
## 421      1 1.153285e-04 1.203461e+01
## 422      1 8.193782e-05 8.736470e+00
## 423      1 8.226990e-05 9.029262e+00
## 424      1 7.446633e-05 8.241831e+00
## 425      1 6.774742e-05 7.566203e+00
## 426      1 4.401149e-05 5.037690e+00
## 427      1 1.643209e-05 2.105690e+00
## 428      1 3.540480e-04 3.190811e+01
## 429      1 5.656922e-04 4.720499e+01
## 430      1 6.503650e-04 5.292099e+01
## 431      1 9.062756e-04 6.970812e+01
## 432      1 9.968572e-04 7.551698e+01
## 433      1 1.447198e-03 1.037618e+02
## 434      1 1.581396e-03 1.120417e+02
## 435      1 1.723003e-03 1.208125e+02
## 436      1 1.796044e-03 1.253117e+02
## 437      1 2.020764e-03 1.393408e+02
## 438      1 2.170321e-03 1.490284e+02
## 439      1 2.358186e-03 1.604812e+02
## 440      1 2.465530e-03 1.672859e+02
## 441      1 2.602748e-03 1.758845e+02
## 442      1 2.729344e-03 1.838535e+02
## 443      1 2.812792e-03 1.914237e+02
## 444      1 2.944173e-03 1.971992e+02
## 445      1 3.039765e-03 2.022694e+02
## 446      1 3.093492e-03 2.062718e+02
## 447      1 3.086716e-03 2.084499e+02
## 448      1 3.148940e-03 2.089634e+02
## 449      1 3.089590e-03 2.094583e+02
## 450      1 3.097804e-03 2.098382e+02
## 451      1 3.101492e-03 2.101391e+02
```

```
## 452      1 3.157670e-03 2.103624e+02
## 453      1 3.107335e-03 2.105012e+02
## 454      1 3.118641e-03 2.105627e+02
## 455      1 3.110130e-03 2.105443e+02
## 456      1 3.163222e-03 2.104348e+02
## 457      1 3.174822e-03 2.102415e+02
## 458      1 3.113529e-03 2.100075e+02
## 459      1 3.108072e-03 2.096716e+02
## 460      1 3.139073e-03 2.092085e+02
## 461      1 3.085225e-03 2.087828e+02
## 462      1 3.082742e-03 2.082069e+02
## 463      1 3.084913e-03 2.067767e+02
## 464      1 3.038732e-03 2.051998e+02
## 465      1 3.014647e-03 2.043247e+02
## 466      1 3.025677e-03 2.031671e+02
## 467      1 2.982253e-03 2.022852e+02
## 468      1 2.956000e-03 2.012395e+02
## 469      1 2.934273e-03 1.986635e+02
## 470      1 2.903125e-03 1.974818e+02
## 471      1 2.850770e-03 1.948527e+02
## 472      1 2.841144e-03 1.931841e+02
## 473      1 2.732136e-03 1.870481e+02
## 474      1 2.616253e-03 1.803003e+02
## 475      1 2.494788e-03 1.730814e+02
## 476      1 2.363787e-03 1.659612e+02
## 477      1 2.238617e-03 1.581716e+02
## 478      1 2.110175e-03 1.507160e+02
## 479      1 1.985780e-03 1.428917e+02
## 480      1 1.863545e-03 1.352060e+02
## 481      1 1.745443e-03 1.285150e+02
## 482      1 1.631929e-03 1.210932e+02
## 483      1 1.525163e-03 1.146632e+02
## 484      1 1.424963e-03 1.085974e+02
## 485      1 1.329293e-03 1.024227e+02
## 486      1 1.240807e-03 9.684105e+01
## 487      1 1.160349e-03 9.195433e+01
## 488      1 1.084955e-03 8.721301e+01
## 489      1 1.014226e-03 8.262121e+01
## 490      1 9.026456e-04 7.659259e+01
## 491      1 8.406383e-04 7.186104e+01
## 492      1 7.904760e-04 6.857851e+01
## 493      1 7.441627e-04 6.551508e+01
## 494      1 7.001795e-04 6.247810e+01
## 495      1 6.624428e-04 6.007050e+01
## 496      1 6.267608e-04 5.770173e+01
## 497      1 5.622040e-04 5.327252e+01
## 498      1 5.073429e-04 4.947898e+01
## 499      1 4.820620e-04 4.764792e+01
## 500      1 4.367659e-04 4.432894e+01
## 501      1 4.013151e-04 4.196772e+01
```

```
## 502      1 3.449344e-04 3.719511e+01
## 503      1 3.171106e-04 3.504673e+01
## 504      1 2.806625e-04 3.211807e+01
## 505      1 2.317122e-04 2.798421e+01
## 506      1 2.154263e-04 2.654956e+01
## 507      1 1.888386e-04 2.424197e+01
## 508      1 1.749503e-04 2.281619e+01
## 509      1 1.695097e-04 2.231376e+01
## 510      1 1.527375e-04 2.059554e+01
## 511      1 1.175836e-04 1.682712e+01
## 512      1 1.097514e-04 1.590160e+01
## 513      1 5.820563e-05 9.401523e+00
## 514      1 2.888537e-05 4.825385e+00
## 515      1 1.297350e-04 1.905952e+01
## 516      1 5.394741e-04 5.661723e+01
## 517      1 6.294905e-04 6.450309e+01
## 518      1 1.473646e-03 1.358219e+02
## 519      1 1.585993e-03 1.455729e+02
## 520      1 1.902033e-03 1.718587e+02
## 521      1 1.930088e-03 1.743105e+02
## 522      1 2.043258e-03 1.836488e+02
## 523      1 2.071334e-03 1.859228e+02
## 524      1 2.098929e-03 1.881983e+02
## 525      1 2.153160e-03 1.926891e+02
## 526      1 2.430420e-03 2.171006e+02
## 527      1 2.509151e-03 2.236158e+02
## 528      1 2.610745e-03 2.289678e+02
## 529      1 2.649401e-03 2.314320e+02
## 530      1 2.666848e-03 2.337471e+02
## 531      1 2.711738e-03 2.374424e+02
## 532      1 2.724270e-03 2.388783e+02
## 533      1 2.741515e-03 2.400190e+02
## 534      1 2.751660e-03 2.408763e+02
## 535      1 2.694036e-03 2.411966e+02
## 536      1 2.761822e-03 2.414448e+02
## 537      1 2.699298e-03 2.416138e+02
## 538      1 2.762387e-03 2.417234e+02
## 539      1 2.708023e-03 2.417546e+02
## 540      1 2.758164e-03 2.417142e+02
## 541      1 2.708677e-03 2.416176e+02
## 542      1 2.751590e-03 2.414214e+02
## 543      1 2.707022e-03 2.412036e+02
## 544      1 2.750937e-03 2.408390e+02
## 545      1 2.698481e-03 2.405200e+02
## 546      1 2.734965e-03 2.399934e+02
## 547      1 2.684315e-03 2.395771e+02
## 548      1 2.722551e-03 2.388658e+02
## 549      1 2.671309e-03 2.383648e+02
## 550      1 2.705551e-03 2.374767e+02
## 551      1 2.678528e-03 2.358730e+02
```

```
## 552      1 2.657207e-03 2.339788e+02
## 553      1 2.630509e-03 2.318665e+02
## 554      1 2.574249e-03 2.298698e+02
## 555      1 2.564143e-03 2.270655e+02
## 556      1 2.492866e-03 2.214398e+02
## 557      1 2.410819e-03 2.152281e+02
## 558      1 2.323661e-03 2.084520e+02
## 559      1 2.230811e-03 2.013031e+02
## 560      1 2.137143e-03 1.936500e+02
## 561      1 2.040110e-03 1.858137e+02
## 562      1 1.941119e-03 1.779593e+02
## 563      1 1.842234e-03 1.701137e+02
## 564      1 1.744628e-03 1.623251e+02
## 565      1 1.649123e-03 1.546132e+02
## 566      1 1.556322e-03 1.471722e+02
## 567      1 1.467305e-03 1.402374e+02
## 568      1 1.380888e-03 1.328689e+02
## 569      1 1.300464e-03 1.266731e+02
## 570      1 1.223366e-03 1.203801e+02
## 571      1 1.151121e-03 1.145522e+02
## 572      1 1.081642e-03 1.086442e+02
## 573      1 1.020080e-03 1.039362e+02
## 574      1 9.615096e-04 9.923045e+01
## 575      1 8.570605e-04 9.085943e+01
## 576      1 8.083743e-04 8.668666e+01
## 577      1 7.655694e-04 8.327292e+01
## 578      1 6.859803e-04 7.652689e+01
## 579      1 6.175215e-04 7.068169e+01
## 580      1 4.401517e-04 5.492041e+01
## 581      1 3.723995e-04 4.881500e+01
## 582      1 3.279485e-04 4.435120e+01
## 583      1 3.019633e-04 4.165065e+01
## 584      1 2.450741e-04 3.604721e+01
## 585      1 2.101999e-04 3.191883e+01
## 586      1 1.752924e-04 2.812044e+01
## 587      1 1.640238e-04 2.713739e+01
## 588      1 1.332237e-04 2.252568e+01
## 589      1 1.311672e-04 2.273090e+01
## 590      1 5.655955e-05 1.147265e+01
## 591      1 3.602589e-05 7.135660e+00
## 592      1 9.094819e-05 8.291975e+01
## 593      1 1.989091e-04 1.813504e+02
## 594      1 2.167979e-04 1.976601e+02
## 595      1 2.234353e-04 2.037116e+02
## 596      1 1.098259e-04 1.807140e+01
## 597      1 1.165998e-04 1.917987e+01
## 598      1 1.073559e-04 1.705187e+01
## 599      1 1.212050e-04 1.805657e+01
## 600      1 1.374397e-04 2.078638e+01
## 601      1 1.497933e-04 2.241653e+01
```

```
## 602      1 1.684270e-04 2.429182e+01
## 603      1 1.955616e-04 2.841461e+01
## 604      1 2.204532e-04 3.082361e+01
## 605      1 2.648177e-04 3.632238e+01
## 606      1 2.979935e-04 4.090037e+01
## 607      1 3.223490e-04 4.363403e+01
## 608      1 3.338710e-04 4.375990e+01
## 609      1 3.560086e-04 4.679594e+01
## 610      1 3.750027e-04 4.920215e+01
## 611      1 3.991865e-04 5.121348e+01
## 612      1 4.178471e-04 5.335413e+01
## 613      1 4.410497e-04 5.629267e+01
## 614      1 5.027457e-04 6.314862e+01
## 615      1 5.306315e-04 6.672256e+01
## 616      1 5.239294e-04 6.338091e+01
## 617      1 5.674170e-04 6.975818e+01
## 618      1 5.908929e-04 7.213027e+01
## 619      1 6.226024e-04 7.609010e+01
## 620      1 6.641007e-04 7.905844e+01
## 621      1 7.106737e-04 8.589732e+01
## 622      1 7.223463e-04 8.495195e+01
## 623      1 8.071459e-04 9.642461e+01
## 624      1 8.748982e-04 1.034718e+02
## 625      1 9.956213e-04 1.182519e+02
## 626      1 1.029336e-03 1.206345e+02
## 627      1 1.064065e-03 1.229056e+02
## 628      1 1.114557e-03 1.310927e+02
## 629      1 1.191940e-03 1.364872e+02
## 630      1 1.237031e-03 1.417823e+02
## 631      1 1.281492e-03 1.463617e+02
## 632      1 1.327478e-03 1.525574e+02
## 633      1 1.418391e-03 1.615717e+02
## 634      1 1.465613e-03 1.654235e+02
## 635      1 1.514053e-03 1.694304e+02
## 636      1 1.558728e-03 1.756773e+02
## 637      1 1.656166e-03 1.845697e+02
## 638      1 1.695242e-03 1.910580e+02
## 639      1 1.756066e-03 1.934477e+02
## 640      1 1.844200e-03 2.038188e+02
## 641      1 1.885754e-03 2.088745e+02
## 642      1 1.935793e-03 2.130006e+02
## 643      1 1.976414e-03 2.177271e+02
## 644      1 2.016930e-03 2.222186e+02
## 645      1 2.099667e-03 2.304936e+02
## 646      1 2.145920e-03 2.341857e+02
## 647      1 2.183232e-03 2.380117e+02
## 648      1 2.209000e-03 2.418984e+02
## 649      1 2.239957e-03 2.453760e+02
## 650      1 2.273408e-03 2.485679e+02
## 651      1 2.301168e-03 2.516162e+02
```

```
## 652      1 2.329836e-03 2.544084e+02
## 653      1 2.352452e-03 2.570161e+02
## 654      1 2.395907e-03 2.614804e+02
## 655      1 2.414967e-03 2.633485e+02
## 656      1 2.427514e-03 2.649795e+02
## 657      1 2.441942e-03 2.663474e+02
## 658      1 2.452583e-03 2.674632e+02
## 659      1 2.462294e-03 2.683221e+02
## 660      1 2.467223e-03 2.689236e+02
## 661      1 2.478896e-03 2.692680e+02
## 662      1 2.473221e-03 2.693527e+02
## 663      1 2.470466e-03 2.691825e+02
## 664      1 2.467707e-03 2.687575e+02
## 665      1 2.461136e-03 2.680825e+02
## 666      1 2.458305e-03 2.671435e+02
## 667      1 2.443280e-03 2.659888e+02
## 668      1 2.425601e-03 2.646104e+02
## 669      1 2.409834e-03 2.629858e+02
## 670      1 2.393176e-03 2.611273e+02
## 671      1 2.352019e-03 2.567817e+02
## 672      1 2.302602e-03 2.516489e+02
## 673      1 2.241049e-03 2.459536e+02
## 674      1 2.177385e-03 2.395704e+02
## 675      1 2.108928e-03 2.326563e+02
## 676      1 2.032967e-03 2.255569e+02
## 677      1 1.956804e-03 2.179787e+02
## 678      1 1.878947e-03 2.101022e+02
## 679      1 1.799493e-03 2.020751e+02
## 680      1 1.720434e-03 1.936475e+02
## 681      1 1.639167e-03 1.858749e+02
## 682      1 1.559665e-03 1.779222e+02
## 683      1 1.481625e-03 1.701926e+02
## 684      1 1.405715e-03 1.626945e+02
## 685      1 1.332003e-03 1.551720e+02
## 686      1 1.260815e-03 1.478935e+02
## 687      1 1.193771e-03 1.414637e+02
## 688      1 1.128682e-03 1.347845e+02
## 689      1 1.068036e-03 1.288507e+02
## 690      1 1.010560e-03 1.231565e+02
## 691      1 9.550966e-04 1.173999e+02
## 692      1 9.025804e-04 1.119076e+02
## 693      1 8.551608e-04 1.072432e+02
## 694      1 8.111089e-04 1.029389e+02
## 695      1 7.688813e-04 9.863999e+01
## 696      1 7.299161e-04 9.472894e+01
## 697      1 6.933892e-04 9.102906e+01
## 698      1 6.592272e-04 8.754731e+01
## 699      1 6.273727e-04 8.429228e+01
## 700      1 5.968140e-04 8.105723e+01
## 701      1 3.810417e-04 5.771440e+01
```

```
## 702     1 2.669116e-04 4.496700e+01
## 703     1 2.460593e-04 4.202491e+01
## 704     1 2.278606e-04 3.948107e+01
## 705     1 1.929836e-04 3.542320e+01
## 706     1 1.714197e-04 3.246267e+01
## 707     1 1.609560e-04 3.089192e+01
## 708     1 1.500945e-04 2.912280e+01
## 709     1 1.389752e-04 2.719745e+01
## 710     1 1.197623e-04 2.447677e+01
## 711     1 1.010180e-04 2.084502e+01
## 712     1 8.044041e-05 1.802478e+01
## 713     1 6.917307e-05 1.593405e+01
## 714     1 6.320715e-05 1.483744e+01
## 715     1 5.346120e-05 1.259330e+01
```