FE621 assignment3 question2

Tengxi Wan

2017-03-23

Qestion 2

(a) Download Data and Calculate Implied Volatility

In this question, we download the option data from Bloomberg Terminal in .csv files. We will use the bisection method to calculate implied volatility with short term interest rate = 0.75%.

```
library(quantmod)

# Set risk free interest rate
r.riskfree = 0.0075

# I download option price data from Bloomberg terminmal on March-01-2017

GS.1m.call=read.csv('/Users/tony/Desktop/515R/GS 030117 1m call.csv')
GS.1m.put=read.csv('/Users/tony/Desktop/515R/GS 030117 1m put.csv')
# one month (17 days)

GS.2m.call=read.csv('/Users/tony/Desktop/515R/GS 030117 2m call.csv')
GS.2m.put=read.csv('/Users/tony/Desktop/515R/GS 030117 2m put.csv')
# two month (52 days)

GS.3m.call=read.csv('/Users/tony/Desktop/515R/GS 030117 3m call.csv')
GS.3m.put=read.csv('/Users/tony/Desktop/515R/GS 030117 3m put.csv')
# three month (77 days)

# For equity price, we use stock price nearly two month ago as the initial price.
getSymbols(Symbols = 'GS', src = 'yahoo', from = '2017-01-06',to = '2017-01-06',auto.assign = TRUE)
```

```
## [1] "GS"
```

```
GS.stock.price =  as.numeric(GS$GS.Adjusted)
```

We will use the bisection method to calculate implied volatility. Because this part of code is very long, and almost the same as in assignment 1. So we only give the result of implied vol while hiding detailed code on this report.

```
bisec.vol.table
```

```
##         1m call     1m put   2m call    2m put   3m call    3m put
##  [1,] 0.8938904 0.22921753 0.7990417 0.2944641 0.7634583 0.3497009
##  [2,] 0.8406677 0.21255493 0.7463074 0.2768860 0.7244568 0.3343201
##  [3,] 0.8006287 0.19314575 0.6938782 0.2610168 0.6894836 0.3207703
##  [4,] 0.7215881 0.18099976 0.6335144 0.2414856 0.6385803 0.3070374
##  [5,] 0.7033386 0.16836548 0.5942688 0.2300110 0.6042175 0.2953796
##  [6,] 0.6250305 0.15231323 0.5461121 0.2144470 0.5855408 0.2834778
##  [7,] 0.6104431 0.13467407 0.5016174 0.1995544 0.5360413 0.2715759
##  [8,] 0.5420837 0.11605835 0.4595642 0.1840515 0.5030212 0.2603455
```

```
##  [9,] 0.5003967 0.09402466 0.4225159 0.1686707 0.4742126 0.2489929
## [10,] 0.4634094 0.06216431 0.3901062 0.1511536 0.4480286 0.2372742
```

(b) EFD, IFD and CNFD Pricing European Call and Put

In this part we will use 3 finite difference methods to price European call and put option. As we have already biuld the function (explicit.method, implicit method and crank.nicolson.method)

Convergence Order

In this part, we will use explicit, implicit and crank-nicolson finite difference method to price European call and put option.The function we have already got in question 1. And we have to decide how many steps to get converge to Black-Scholes price, with error magitude at 0.001. Recall that we have convergence condition for EFD, and the error's order is:

$$O(\Delta x^2 + \Delta t), \quad with \quad \Delta x \geq \sigma\sqrt{3\Delta t}$$

So that,we have:

$$N \geq (3\sigma^2 + 1)\frac{T}{\epsilon}$$

And for crank-nicolson method, it is more accuracy and converges faster than EFD and IFD. The error order is

$$O(\Delta x^2 + (\frac{\Delta t}{2})^2)$$

For implicit and crank-nicolson method, they are unconditional stable and convergence. We may not get an unique steps so that it will converge because it has two variables(dx and dt). However, we can take explicit as freference and set dx smaller than in explicit method and use the same steps to converge as in EFD to make it accurate.

Finite Difference Method Pricing

Here we use EFD,IFD and CNFD method to price 1m,2m and 3m european call and put option with the parameters we get before. The result we will show in the following full table.

```
explicit.price.1m.call = c()
explicit.dx.1m.call=c()
explicit.N.1m.call = c()
explicit.dt.1m.call = c()
# these are to get N(steps),dt and dx and price
for(i in 1:10){
    explicit.N.1m.call[i] = ceiling((3*bisec.vol.1m.call[i]^2+1)*1000*(17/360))
    explicit.dt.1m.call[i] = (17/360)/explicit.N.1m.call[i]
    explicit.dx.1m.call[i] = bisec.vol.1m.call[i]*sqrt(3*explicit.dt.1m.call[i])
    explicit.price.1m.call[i] = explicit.method(S = GS.stock.price,K = GS.1m.call.strike[i],tao = 17/360
                                r= r.riskfree,sigma = bisec.vol.1m.call[i],step = explicit.N.1m.call[i]
                                div = 0, dx = explicit.dx.1m.call[i],first = 3,type1 = 'European',
                                type2 = 'Call')$Price
}

bisec.vol.1m.put
```

```
##  [1] 0.22921753 0.21255493 0.19314575 0.18099976 0.16836548 0.15231323
##  [7] 0.13467407 0.11605835 0.09402466 0.06216431
```

```
explicit.price.1m.put = c()
explicit.dx.1m.put=c()
explicit.N.1m.put = c()
explicit.dt.1m.put = c()
for(i in 1:10){
    explicit.N.1m.put[i] = ceiling((3*bisec.vol.1m.put[i]^2+1)*1000*(17/360))
    explicit.dt.1m.put[i] = (17/360)/explicit.N.1m.put[i]
    explicit.dx.1m.put[i] = bisec.vol.1m.put[i]*sqrt(3*explicit.dt.1m.put[i])
    explicit.price.1m.put[i] = explicit.method(S = GS.stock.price,K = GS.1m.put.strike[i],tao = 17/360,
                               r= r.riskfree,sigma = bisec.vol.1m.put[i],step = explicit.N.1m.put[i],
                               div = 0, dx = explicit.dx.1m.put[i],first = 3,type1 = 'European',
                               type2 = 'Put')$Price
}

explicit.price.2m.call = c()
explicit.dx.2m.call=c()
explicit.N.2m.call = c()
explicit.dt.2m.call = c()
for(i in 1:10){
    explicit.N.2m.call[i] = ceiling((3*bisec.vol.2m.call[i]^2+1)*1000*(52/360))
    explicit.dt.2m.call[i] = (52/360)/explicit.N.2m.call[i]
    explicit.dx.2m.call[i] = bisec.vol.2m.call[i]*sqrt(3*explicit.dt.2m.call[i])
    explicit.price.2m.call[i] = explicit.method(S = GS.stock.price,K = GS.2m.call.strike[i],tao = 52/360
                                r= r.riskfree,sigma = bisec.vol.2m.call[i],step = explicit.N.2m.call[i]
                                div = 0,dx = explicit.dx.2m.call[i],first = 3,type1 = 'European',
                                type2 = 'Call')$Price
}

explicit.price.2m.put = c()
explicit.dx.2m.put=c()
explicit.N.2m.put = c()
explicit.dt.2m.put = c()
for(i in 1:10){
    explicit.N.2m.put[i] = ceiling((3*bisec.vol.2m.put[i]^2+1)*1000*(52/360))
    explicit.dt.2m.put[i] = (52/360)/explicit.N.2m.put[i]
    explicit.dx.2m.put[i] = bisec.vol.2m.put[i]*sqrt(3*explicit.dt.2m.put[i])
    explicit.price.2m.put[i] = explicit.method(S = GS.stock.price,K = GS.2m.put.strike[i],tao = 52/360,
                               r= r.riskfree,sigma = bisec.vol.2m.put[i],step = explicit.N.2m.put[i],
                               div = 0,dx = explicit.dx.2m.put[i],first = 3,type1 = 'European',
                               type2 = 'Put')$Price
}

explicit.price.3m.call = c()
explicit.dx.3m.call=c()
explicit.N.3m.call = c()
explicit.dt.3m.call = c()
for(i in 1:10){
    explicit.N.3m.call[i] = ceiling((3*bisec.vol.3m.call[i]^2+1)*1000*(77/360))
    explicit.dt.3m.call[i] = (77/360)/explicit.N.3m.call[i]
    explicit.dx.3m.call[i] = bisec.vol.3m.call[i]*sqrt(3*explicit.dt.3m.call[i])
    explicit.price.3m.call[i] = explicit.method(S = GS.stock.price,K = GS.3m.call.strike[i],tao = 77/360
```

```
                                r= r.riskfree,sigma = bisec.vol.3m.call[i],step = explicit.N.3m.call[i]
                                div = 0,dx = explicit.dx.3m.call[i],first = 3,type1 = 'European',
                                type2 = 'Call')$Price
}


explicit.price.3m.put = c()
explicit.dx.3m.put=c()
explicit.N.3m.put = c()
explicit.dt.3m.put = c()
for(i in 1:10){
    explicit.N.3m.put[i] = ceiling((3*bisec.vol.3m.put[i]^2+1)*1000*(77/360))
    explicit.dt.3m.put[i] = (77/360)/explicit.N.3m.put[i]
    explicit.dx.3m.put[i] = bisec.vol.3m.put[i]*sqrt(3*explicit.dt.1m.put[i])
    explicit.price.3m.put[i] = explicit.method(S = GS.stock.price,K = GS.3m.put.strike[i],tao = 77/360,
                               r= r.riskfree,sigma = bisec.vol.3m.put[i],step = explicit.N.3m.put[i],
                               div = 0,dx = explicit.dx.3m.put[i],first = 3,type1 = 'European',
                               type2 = 'Put')$Price
}

implicit.price.1m.call = c()
implicit.dx.1m.call=c()
implicit.dt.1m.call = c()
for(i in 1:10){
    implicit.dt.1m.call[i] = (17/360)/explicit.N.1m.call[i]
    # Note implicit has same converge order, we can use explicit method steps here
    implicit.price.1m.call[i] = implicit.method(S = GS.stock.price,K = GS.1m.call.strike[i],tao = 17/360
                                r= r.riskfree,sigma = bisec.vol.1m.call[i], step = explicit.N.1m.call[i]
                                div = 0, dx = 0.02,first = 3,type1 = 'European',type2 = 'Call')$Price
}
# Note that we use explicit steps and smaller dx = 0.02

implicit.price.1m.put = c()
implicit.dx.1m.put=c()
implicit.dt.1m.put = c()
for(i in 1:10){
    implicit.dt.1m.put[i] = (17/360)/explicit.N.1m.put[i]
    implicit.price.1m.put[i] = implicit.method(S = GS.stock.price,K = GS.1m.put.strike[i],tao = 17/360,
                               r= r.riskfree,sigma = bisec.vol.1m.put[i],step = explicit.N.1m.put[i],
                               div = 0, dx = 0.02,first = 3,type1 = 'European',type2 = 'Put')$Price
}

implicit.price.2m.call = c()
implicit.dx.2m.call=c()
implicit.dt.2m.call = c()
for(i in 1:10){
    implicit.dt.2m.call[i] = (52/360)/explicit.N.2m.call[i]
    implicit.price.2m.call[i] = implicit.method(S = GS.stock.price,K = GS.2m.call.strike[i],tao = 52/360
                                r= r.riskfree,sigma = bisec.vol.2m.call[i],step =explicit.N.2m.call[i],
                                div = 0,dx = 0.02,first = 3,type1 = 'European',type2 = 'Call')$Price
}

implicit.price.2m.put = c()
```

```
implicit.dx.2m.put=c()
implicit.dt.2m.put = c()
for(i in 1:10){
    implicit.dt.2m.put[i] = (52/360)/explicit.N.2m.put[i]
    implicit.price.2m.put[i] = implicit.method(S = GS.stock.price,K = GS.2m.put.strike[i],tao = 52/360,
                               r= r.riskfree,sigma = bisec.vol.2m.put[i],step = explicit.N.2m.put[i],
                               div = 0, dx = 0.02,first = 3,type1 = 'European',type2 = 'Put')$Price
}

implicit.price.3m.call = c()
implicit.dx.3m.call=c()
implicit.dt.3m.call = c()
for(i in 1:10){
    implicit.dt.3m.call[i] = (77/360)/explicit.N.3m.call[i]
    implicit.price.3m.call[i] = implicit.method(S = GS.stock.price,K = GS.3m.call.strike[i],tao = 77/360
                               r= r.riskfree,sigma = bisec.vol.3m.call[i],step = explicit.N.3m.call[i],
                               div = 0, dx = 0.02,first = 3,type1 = 'European',type2 = 'Call')$Price
}


implicit.price.3m.put = c()
implicit.dx.3m.put=c()
implicit.dt.3m.put = c()
for(i in 1:10){
    implicit.dt.3m.put[i] = (77/360)/explicit.N.3m.put[i]
    implicit.price.3m.put[i] = implicit.method(S = GS.stock.price,K = GS.3m.put.strike[i],tao = 77/360,
                               r= r.riskfree,sigma = bisec.vol.3m.put[i],step = explicit.N.3m.put[i],
                               div = 0,dx = 0.02,first = 3,type1 = 'European',type2 = 'Put')$Price
}

crank.nicolson.price.1m.call = c()
crank.nicolson.dx.1m.call=c()
crank.nicolson.dt.1m.call = c()
for(i in 1:10){
    crank.nicolson.dt.1m.call[i] = (17/360)/explicit.N.1m.call[i]
    # Note crank nicolson has better converge order, so we can use the explicit method steps
    crank.nicolson.price.1m.call[i] = crank.nicolson.method(S = GS.stock.price,K = GS.1m.call.strike[i]
                                       tao = 17/360,r= r.riskfree,sigma = bisec.vol.1m.call[i],
                                      step = explicit.N.1m.call[i], div = 0,dx = 0.02,first = 1,
                                      type1 = 'European',type2 = 'Call')$Price
}

crank.nicolson.price.1m.put = c()
crank.nicolson.dx.1m.put=c()
crank.nicolson.dt.1m.put = c()
for(i in 1:10){
    crank.nicolson.dt.1m.put[i] = (17/360)/explicit.N.1m.put[i]
    crank.nicolson.price.1m.put[i] = crank.nicolson.method(S = GS.stock.price,K = GS.1m.put.strike[i],
                                      tao = 17/360,r= r.riskfree,sigma = bisec.vol.1m.put[i],
                                      step = explicit.N.1m.put[i], div = 0, dx = 0.02,first = 1,
                                      type1 = 'European',type2 = 'Put')$Price
}
```

```
crank.nicolson.price.2m.call = c()
crank.nicolson.dx.2m.call=c()
crank.nicolson.dt.2m.call = c()
for(i in 1:10){
    crank.nicolson.dt.2m.call[i] = (52/360)/explicit.N.2m.call[i]
    crank.nicolson.price.2m.call[i] = crank.nicolson.method(S = GS.stock.price,K = GS.2m.call.strike[i]
                                      tao=52/360,r= r.riskfree,sigma = bisec.vol.2m.call[i],
                                      step = explicit.N.2m.call[i], div = 0,dx = 0.02,first = 1,
                                      type1 = 'European',type2 = 'Call')$Price
}

crank.nicolson.price.2m.put = c()
crank.nicolson.dx.2m.put=c()
crank.nicolson.dt.2m.put = c()
for(i in 1:10){
    crank.nicolson.dt.2m.put[i] = (52/360)/explicit.N.2m.put[i]
    crank.nicolson.price.2m.put[i] = crank.nicolson.method(S = GS.stock.price,K = GS.2m.put.strike[i],
                                     tao=52/360,r= r.riskfree,sigma = bisec.vol.2m.put[i],
                                     step = explicit.N.2m.put[i], div = 0,dx = 0.02,first = 1,
                                     type1 = 'European',type2 = 'Put')$Price
}

crank.nicolson.price.3m.call = c()
crank.nicolson.dx.3m.call=c()
crank.nicolson.dt.3m.call = c()
for(i in 1:10){
    crank.nicolson.dt.3m.call[i] = (77/360)/explicit.N.3m.call[i]
    crank.nicolson.price.3m.call[i] = crank.nicolson.method(S = GS.stock.price,K = GS.3m.call.strike[i]
                                      tao=77/360,r= r.riskfree,sigma = bisec.vol.3m.call[i],
                                      step = explicit.N.3m.call[i], div = 0,dx = 0.02,first = 1,
                                      type1 = 'European',type2 = 'Call')$Price
}

crank.nicolson.price.3m.put = c()
crank.nicolson.dx.3m.put=c()
crank.nicolson.dt.3m.put = c()
for(i in 1:10){
    crank.nicolson.dt.3m.put[i] = (77/360)/explicit.N.3m.put[i]
    crank.nicolson.price.3m.put[i] = crank.nicolson.method(S = GS.stock.price,K = GS.3m.put.strike[i],
                                     tao=77/360,r= r.riskfree,sigma = bisec.vol.3m.put[i],
                                     step = explicit.N.3m.put[i], div = 0,dx = 0.02,first = 1,
                                      type1 = 'European',type2 = 'Put')$Price
}
```

(c) Greeks by Explicit Method

Greeks 1 month call

```
explicit.delta.1m.call = c()
explicit.gamma.1m.call = c()
explicit.vega.1m.call =c()
explicit.theta.1m.call = c()
```

```
for(i in 1:10){
    explicit.method.1m.call = explicit.method(S=GS.stock.price,K=GS.1m.call.strike[i],tao=17/360,
                                              r=r.riskfree,sigma=bisec.vol.1m.call[i],
                                              div=0,step=explicit.N.1m.call[i],dx=explicit.dx.1m.call[i],
                                              first=3, type1='Eourpean',type2='Call')
    explicit.method.1m.call.first.v = as.numeric(explicit.method.1m.call$V.first.steps[,1])
    explicit.method.1m.call.first.v.plus = as.numeric(explicit.method.1m.call$V.first.steps[,2])
    explicit.method.1m.call.first.s = as.numeric(explicit.method.1m.call$S.first.steps[,4])
    explicit.delta.1m.call[i] = ((explicit.method.1m.call.first.v[3]-explicit.method.1m.call.first.v[5])
                                 (explicit.method.1m.call.first.s[3]-explicit.method.1m.call.first.s[5]))

    explicit.gamma.1m.call[i]= ((explicit.method.1m.call.first.v[3]-explicit.method.1m.call.first.v[4])-
                                  (explicit.method.1m.call.first.v[4]-explicit.method.1m.call.first.v[5])
                                  (explicit.method.1m.call.first.s[3]-explicit.method.1m.call.first.s[5]),

    price.1m.call.sigma = explicit.method(S=GS.stock.price,K=GS.1m.call.strike[i],tao=17/360,
                                          r=r.riskfree,sigma=(bisec.vol.1m.call[i]+0.001),
                                          div=0,step=explicit.N.1m.call[i],dx=explicit.dx.1m.call[i],
                                          first=3, type1='Eourpean',type2='Call')$Price
    price.1m.call.sigma.plus = explicit.method(S=GS.stock.price,K=GS.1m.call.strike[i],tao=17/360,
                                               r=r.riskfree,sigma=(bisec.vol.1m.call[i]-0.001),
                                               div=0,step=explicit.N.1m.call[i],dx=explicit.dx.1m.call[
                                               first=3, type1='Eourpean',type2='Call')$Price
    explicit.vega.1m.call[i] = (price.1m.call.sigma.plus-price.1m.call.sigma)/(2*0.001)

    explicit.theta.1m.call[i] = (explicit.method.1m.call.first.v.plus[4]-explicit.method.1m.call.first.v
}

greeks.1m.call = cbind(GS.1m.call.strike,explicit.delta.1m.call,explicit.gamma.1m.call,
                       explicit.vega.1m.call,explicit.theta.1m.call)
colnames(greeks.1m.call) = c('Strike','Delta','Gamma','Vega','Theta')
greeks.1m.call
```

```
##       Strike     Delta      Gamma      Vega     Theta
##  [1,]  222.5 0.7185185 0.01625134 -17.91420 -59.45822
##  [2,]  225.0 0.7062289 0.01683145 -18.30415 -62.16884
##  [3,]  227.5 0.6904501 0.01733502 -18.73534 -64.53286
##  [4,]  230.0 0.6784574 0.01824883 -19.02914 -67.90435
##  [5,]  232.5 0.6560298 0.01866152 -19.60417 -69.90572
##  [6,]  235.0 0.6385531 0.01963141 -19.91004 -72.33020
##  [7,]  237.5 0.6104320 0.01997208 -20.40467 -73.80864
##  [8,]  240.0 0.5837817 0.02101617 -20.76957 -75.75338
##  [9,]  242.5 0.5496014 0.02157810 -21.05326 -76.01750
## [10,]  245.0 0.5100042 0.02202255 -21.25357 -75.62562
```

```
# This is the greeks table for 1 month call option
```

Greeks 1 month put

```
explicit.delta.1m.put = c()
explicit.gamma.1m.put = c()
explicit.vega.1m.put =c()
explicit.theta.1m.put = c()
```

```
for(i in 1:10){
    explicit.method.1m.put = explicit.method(S=GS.stock.price,K=GS.1m.put.strike[i],tao=17/360,
                                             r=r.riskfree,sigma=bisec.vol.1m.put[i],
                                             div=0,step=explicit.N.1m.put[i],dx=explicit.dx.1m.put[i],
                                             first=3, type1='Eourpean',type2='Put')
    explicit.method.1m.put.first.v = as.numeric(explicit.method.1m.put$V.first.steps[,1])
    explicit.method.1m.put.first.v.plus = as.numeric(explicit.method.1m.put$V.first.steps[,2])
    explicit.method.1m.put.first.s = as.numeric(explicit.method.1m.put$S.first.steps[,4])
    explicit.delta.1m.put[i] = ((explicit.method.1m.put.first.v[3]-explicit.method.1m.put.first.v[5])/
                                    (explicit.method.1m.put.first.s[3]-explicit.method.1m.put.first.s[5]

    explicit.gamma.1m.put[i]= ((explicit.method.1m.put.first.v[3]-explicit.method.1m.put.first.v[4])-
                                   (explicit.method.1m.put.first.v[4]-explicit.method.1m.put.first.v[5])
        (explicit.method.1m.put.first.s[3]-explicit.method.1m.put.first.s[5])/2

    price.1m.put.sigma = explicit.method(S=GS.stock.price,K=GS.1m.put.strike[i],tao=17/360,
                                         r=r.riskfree,sigma=(bisec.vol.1m.put[i]+0.001),
                                         div=0,step=explicit.N.1m.put[i],dx=explicit.dx.1m.put[i],
                                         first=3, type1='Eourpean',type2='put')$Price
    price.1m.put.sigma.plus = explicit.method(S=GS.stock.price,K=GS.1m.put.strike[i],tao=17/360,
                                              r=r.riskfree,sigma=(bisec.vol.1m.put[i]-0.001),
                                              div=0,step=explicit.N.1m.put[i],dx=explicit.dx.1m.put[i],
                                              first=3, type1='Eourpean',type2='put')$Price
    explicit.vega.1m.put[i] = (price.1m.put.sigma.plus-price.1m.put.sigma)/(2*0.001)

    explicit.theta.1m.put[i] = (explicit.method.1m.put.first.v.plus[4]-explicit.method.1m.put.first.v[4]
}

greeks.1m.put = cbind(GS.1m.put.strike,explicit.delta.1m.put,explicit.gamma.1m.put,
                      explicit.vega.1m.put,explicit.theta.1m.put)
colnames(greeks.1m.put) = c('Strike','Delta','Gamma','Vega','Theta')
greeks.1m.put
```

```
##       Strike       Delta       Gamma       Vega      Theta
##  [1,]  222.5 -0.02894882 0.003713149  -3.372908  -8.296719
##  [2,]  225.0 -0.03677507 0.004545228  -4.102609  -9.523499
##  [3,]  227.5 -0.04353436 0.005297652  -4.739661 -10.176887
##  [4,]  230.0 -0.06159549 0.007034796  -6.266737 -12.836710
##  [5,]  232.5 -0.08665032 0.009145338  -8.188484 -15.578549
##  [6,]  235.0 -0.11808363 0.011620583 -10.343023 -18.111536
##  [7,]  237.5 -0.16370832 0.014724627 -13.027025 -20.505837
##  [8,]  240.0 -0.23613690 0.018382424 -16.333065 -22.031758
##  [9,]  242.5 -0.35292767 0.022352076 -19.737358 -21.749144
## [10,]  245.0 -0.57490955 0.023929502 -20.942192 -14.862873
```

```
# This is the greeks table for 1 month put option
```

Greeks 2 months call

```
explicit.delta.2m.call = c()
explicit.gamma.2m.call = c()
explicit.vega.2m.call =c()
explicit.theta.2m.call = c()
```

```
for(i in 1:10){
    explicit.method.2m.call = explicit.method(S=GS.stock.price,K=GS.2m.call.strike[i],tao=52/360,
                                              r=r.riskfree,sigma=bisec.vol.2m.call[i],
                                              div=0,step=explicit.N.2m.call[i],dx=explicit.dx.2m.call[i]
                                              first=3, type1='Eourpean',type2='Call')
    explicit.method.2m.call.first.v = as.numeric(explicit.method.2m.call$V.first.steps[,1])
    explicit.method.2m.call.first.v.plus = as.numeric(explicit.method.2m.call$V.first.steps[,2])
    explicit.method.2m.call.first.s = as.numeric(explicit.method.2m.call$S.first.steps[,4])
    explicit.delta.2m.call[i] = ((explicit.method.2m.call.first.v[3]-explicit.method.2m.call.first.v[5]
                                  (explicit.method.2m.call.first.s[3]-explicit.method.2m.call.first.s

    explicit.gamma.2m.call[i]= ((explicit.method.2m.call.first.v[3]-explicit.method.2m.call.first.v[4])-
                                 (explicit.method.2m.call.first.v[4]-explicit.method.2m.call.first.v
        (explicit.method.2m.call.first.s[3]-explicit.method.2m.call.first.s[5])/2

    price.2m.call.sigma = explicit.method(S=GS.stock.price,K=GS.2m.call.strike[i],tao=52/360,
                                          r=r.riskfree,sigma=(bisec.vol.2m.call[i]+0.001),
                                          div=0,step=explicit.N.2m.call[i],dx=explicit.dx.2m.call[i],
                                          first=3, type1='Eourpean',type2='Call')$Price
    price.2m.call.sigma.plus = explicit.method(S=GS.stock.price,K=GS.2m.call.strike[i],tao=52/360,
                                               r=r.riskfree,sigma=(bisec.vol.2m.call[i]-0.001),
                                               div=0,step=explicit.N.2m.call[i],dx=explicit.dx.2m.call[
                                               first=3, type1='Eourpean',type2='Call')$Price
    explicit.vega.2m.call[i] = (price.2m.call.sigma.plus-price.2m.call.sigma)/(2*0.001)

    explicit.theta.2m.call[i] =(explicit.method.2m.call.first.v.plus[4]-
                                explicit.method.2m.call.first.v[4])/(1/1188)
}
greeks.2m.call = cbind(GS.2m.call.strike,explicit.delta.2m.call,explicit.gamma.2m.call,
                       explicit.vega.2m.call,explicit.theta.2m.call)
colnames(greeks.2m.call) = c('Strike','Delta','Gamma','Vega','Theta')
greeks.2m.call
```

```
##       Strike     Delta      Gamma      Vega     Theta
##  [1,]    195 0.8150809 0.01083815 -24.77061 -28.29000
##  [2,]    200 0.8024692 0.01113955 -25.81096 -30.11929
##  [3,]    205 0.7881541 0.01144452 -26.89337 -31.83665
##  [4,]    210 0.7739626 0.01179818 -27.90702 -33.50938
##  [5,]    215 0.7524318 0.01218197 -29.34518 -35.38818
##  [6,]    220 0.7299667 0.01260509 -30.71262 -37.03991
##  [7,]    225 0.7025792 0.01304608 -32.19626 -38.49064
##  [8,]    230 0.6692951 0.01350768 -33.70136 -39.73949
##  [9,]    235 0.6284821 0.01391533 -35.15859 -40.51866
## [10,]    240 0.5794312 0.01421626 -36.38460 -40.72005
```

```
# This is the greeks table for 2 months call option
```

Greeks 2 months put

```
explicit.delta.2m.put = c()
explicit.gamma.2m.put = c()
explicit.vega.2m.put =c()
explicit.theta.2m.put = c()
```

```
for(i in 1:10){
    explicit.method.2m.put = explicit.method(S=GS.stock.price,K=GS.2m.put.strike[i],tao=52/360,
                                             r=r.riskfree,sigma=bisec.vol.2m.put[i],
                                             div=0,step=explicit.N.2m.put[i],dx=explicit.dx.2m.put[i],
                                             first=3, type1='Eourpean',type2='Put')
    explicit.method.2m.put.first.v = as.numeric(explicit.method.2m.put$V.first.steps[,1])
    explicit.method.2m.put.first.v.plus = as.numeric(explicit.method.2m.put$V.first.steps[,2])
    explicit.method.2m.put.first.s = as.numeric(explicit.method.2m.put$S.first.steps[,4])
    explicit.delta.2m.put[i] = ((explicit.method.2m.put.first.v[3]-explicit.method.2m.put.first.v[5])/
                                    (explicit.method.2m.put.first.s[3]-explicit.method.2m.put.first.s[5]

    explicit.gamma.2m.put[i]= ((explicit.method.2m.put.first.v[3]-explicit.method.2m.put.first.v[4])-
                                   (explicit.method.2m.put.first.v[4]-explicit.method.2m.put.first.v[5])
        (explicit.method.2m.put.first.s[3]-explicit.method.2m.put.first.s[5])/2

    price.2m.put.sigma = explicit.method(S=GS.stock.price,K=GS.2m.put.strike[i],tao=52/360,
                                         r=r.riskfree,sigma=(bisec.vol.2m.put[i]+0.001),
                                         div=0,step=explicit.N.2m.put[i],dx=explicit.dx.2m.put[i],
                                         first=3, type1='Eourpean',type2='Put')$Price
    price.2m.put.sigma.plus = explicit.method(S=GS.stock.price,K=GS.2m.put.strike[i],tao=52/360,
                                              r=r.riskfree,sigma=(bisec.vol.2m.put[i]-0.001),
                                              div=0,step=explicit.N.2m.put[i],dx=explicit.dx.2m.put[i],
                                              first=3, type1='Eourpean',type2='Put')$Price
    explicit.vega.2m.put[i] = (price.2m.put.sigma.plus-price.2m.put.sigma)/(2*0.001)

    explicit.theta.2m.put[i] = (explicit.method.2m.put.first.v.plus[4]-explicit.method.2m.put.first.v[4]
}

greeks.2m.put = cbind(GS.2m.put.strike,explicit.delta.2m.put,explicit.gamma.2m.put,
                      explicit.vega.2m.put,explicit.theta.2m.put)
colnames(greeks.2m.put) = c('Strike','Delta','Gamma','Vega','Theta')
greeks.2m.put
```

```
##       Strike       Delta       Gamma       Vega      Theta
##  [1,]    195 -0.01910644 0.001412391  -4.259386  -4.037339
##  [2,]    200 -0.02518857 0.001810335  -5.384970  -4.929893
##  [3,]    205 -0.03429729 0.002367736  -6.969650  -6.146601
##  [4,]    210 -0.04466446 0.002975327  -8.654251  -7.217358
##  [5,]    215 -0.06514112 0.004037714 -11.709345  -9.397099
##  [6,]    220 -0.09124969 0.005254979 -15.132816 -11.503375
##  [7,]    225 -0.12851474 0.006793821 -19.433174 -13.961674
##  [8,]    230 -0.18198699 0.008583223 -24.478465 -16.355905
##  [9,]    235 -0.25787970 0.010561109 -29.966088 -18.587459
## [10,]    240 -0.36198068 0.012291746 -34.841069 -19.418984
```

```
# This is the greeks table for 2 months put option
```

Greeks 3 months call

```
explicit.delta.3m.call = c()
explicit.gamma.3m.call = c()
explicit.vega.3m.call =c()
explicit.theta.3m.call = c()
```

```
for(i in 1:10){
    explicit.method.3m.call = explicit.method(S=GS.stock.price,K=GS.3m.call.strike[i],tao=77/360,
                                              r=r.riskfree,sigma=bisec.vol.3m.call[i],
                                              div=0,step=explicit.N.3m.call[i],dx=explicit.dx.3m.call[i]
                                              first=3, type1='Eourpean',type2='Call')
    explicit.method.3m.call.first.v = as.numeric(explicit.method.3m.call$V.first.steps[,1])
    explicit.method.3m.call.first.v.plus = as.numeric(explicit.method.3m.call$V.first.steps[,2])
    explicit.method.3m.call.first.s = as.numeric(explicit.method.3m.call$S.first.steps[,4])
    explicit.delta.3m.call[i] = ((explicit.method.3m.call.first.v[3]-explicit.method.3m.call.first.v[5]
                                    (explicit.method.3m.call.first.s[3]-explicit.method.3m.call.first.s

    explicit.gamma.3m.call[i]= ((explicit.method.3m.call.first.v[3]-explicit.method.3m.call.first.v[4])-
                                   (explicit.method.3m.call.first.v[4]-explicit.method.3m.call.first.v
        (explicit.method.3m.call.first.s[3]-explicit.method.3m.call.first.s[5])/2

    price.3m.call.sigma = explicit.method(S=GS.stock.price,K=GS.3m.call.strike[i],tao=77/360,
                                          r=r.riskfree,sigma=(bisec.vol.3m.call[i]+0.001),
                                          div=0,step=explicit.N.3m.call[i],dx=explicit.dx.3m.call[i],
                                          first=3, type1='Eourpean',type2='Call')$Price
    price.3m.call.sigma.plus = explicit.method(S=GS.stock.price,K=GS.3m.call.strike[i],tao=77/360,
                                               r=r.riskfree,sigma=(bisec.vol.3m.call[i]-0.001),
                                               div=0,step=explicit.N.3m.call[i],dx=explicit.dx.3m.call[
                                               first=3, type1='Eourpean',type2='Call')$Price
    explicit.vega.3m.call[i] = (price.3m.call.sigma.plus-price.3m.call.sigma)/(2*0.001)

    explicit.theta.3m.call[i] = (explicit.method.3m.call.first.v.plus[4]-
                                    explicit.method.3m.call.first.v[4])/(1/1188)
}

greeks.3m.call = cbind(GS.3m.call.strike,explicit.delta.3m.call,explicit.gamma.3m.call,
                       explicit.vega.3m.call,explicit.theta.3m.call)
colnames(greeks.3m.call) = c('Strike','Delta','Gamma','Vega','Theta')
greeks.3m.call
```

```
##       Strike     Delta       Gamma      Vega     Theta
##  [1,]    185 0.8335047 0.009713015 -28.21277 -22.19678
##  [2,]    190 0.8217141 0.009887631 -29.46467 -23.48527
##  [3,]    195 0.8079963 0.010084539 -30.84916 -24.85112
##  [4,]    200 0.7966314 0.010276551 -31.93663 -26.01245
##  [5,]    205 0.7800062 0.010499515 -33.44544 -27.33319
##  [6,]    210 0.7577813 0.010754043 -35.29751 -28.90104
##  [7,]    215 0.7405730 0.011001216 -36.62646 -29.89986
##  [8,]    220 0.7166498 0.011266732 -38.28150 -31.04381
##  [9,]    225 0.6884657 0.011527278 -39.97445 -32.08974
## [10,]    230 0.6560683 0.011778898 -41.61372 -33.02423
```

```
# This is the greeks table for 3 months call option
```

Greeks 3 months put

```
explicit.delta.3m.put = c()
explicit.gamma.3m.put = c()
explicit.vega.3m.put =c()
```

```
explicit.theta.3m.put = c()
for(i in 1:10){
    explicit.method.3m.put = explicit.method(S=GS.stock.price,K=GS.3m.put.strike[i],tao=77/360,
                                             r=r.riskfree,sigma=bisec.vol.3m.put[i],
                                             div=0,step=explicit.N.3m.put[i],dx=explicit.dx.3m.put[i],
                                             first=3, type1='Eourpean',type2='Put')
    explicit.method.3m.put.first.v = as.numeric(explicit.method.3m.put$V.first.steps[,1])
    explicit.method.3m.put.first.v.plus = as.numeric(explicit.method.3m.put$V.first.steps[,2])
    explicit.method.3m.put.first.s = as.numeric(explicit.method.3m.put$S.first.steps[,4])
    explicit.delta.3m.put[i] = ((explicit.method.3m.put.first.v[3]-explicit.method.3m.put.first.v[5])/
                                    (explicit.method.3m.put.first.s[3]-explicit.method.3m.put.first.s[5]

    explicit.gamma.3m.put[i]= ((explicit.method.3m.put.first.v[3]-explicit.method.3m.put.first.v[4])-
                                   (explicit.method.3m.put.first.v[4]-explicit.method.3m.put.first.v[5])
        (explicit.method.3m.put.first.s[3]-explicit.method.3m.put.first.s[5])/2

    price.3m.put.sigma = explicit.method(S=GS.stock.price,K=GS.3m.put.strike[i],tao=77/360,
                                         r=r.riskfree,sigma=(bisec.vol.3m.put[i]+0.001),
                                         div=0,step=explicit.N.3m.put[i],dx=explicit.dx.3m.put[i],
                                         first=3, type1='Eourpean',type2='Put')$Price
    price.3m.put.sigma.plus = explicit.method(S=GS.stock.price,K=GS.3m.put.strike[i],tao=77/360,
                                              r=r.riskfree,sigma=(bisec.vol.3m.put[i]-0.001),
                                              div=0,step=explicit.N.3m.put[i],dx=explicit.dx.3m.put[i],
                                              first=3, type1='Eourpean',type2='Put')$Price
    explicit.vega.3m.put[i] = (price.3m.put.sigma.plus-price.3m.put.sigma)/(2*0.001)

    explicit.theta.3m.put[i] = (explicit.method.3m.put.first.v.plus[4]-explicit.method.3m.put.first.v[4]
}

greeks.3m.put = cbind(GS.3m.put.strike,explicit.delta.3m.put,explicit.gamma.3m.put,
                      explicit.vega.3m.put,explicit.theta.3m.put)
colnames(greeks.3m.put) = c('Strike','Delta','Gamma','Vega','Theta')
head(greeks.3m.put)
```

```
##      Strike       Delta       Gamma       Vega      Theta
## [1,]    185 -0.03550379 0.001978190  -8.765573  -6.154505
## [2,]    190 -0.04371559 0.002366533 -10.390012  -7.138931
## [3,]    195 -0.05480341 0.002861173 -12.455298  -8.379103
## [4,]    200 -0.06849788 0.003438069 -14.838400  -9.717867
## [5,]    205 -0.08690095 0.004115528 -17.798370 -11.406967
## [6,]    210 -0.10952197 0.004921524 -21.124189 -13.168593
```

```
# This is the greeks table for 3 months put option
```

(d) Table and Plot

The Detailed Table

```
maturity.1m = rep(17,10)
maturity.2m = rep(52,10)
maturity.3m = rep(77,10)
type.1m.call = type.2m.call = type.3m.call = rep('Call',10)
```

```
type.1m.put = type.2m.put = type.3m.put = rep('Put',10)
detailed.table.1m.call = cbind(maturity.1m,type.1m.call,GS.1m.call.strike,round(GS.1m.call.bid,3),
                               round(GS.1m.call.ask,3),round(market.price.1m.call,3),
                               round(explicit.price.1m.call,3),round(implicit.price.1m.call,3),
                               round(crank.nicolson.price.1m.call,3),round(bisec.vol.1m.call,3))
detailed.table.1m.put = cbind(maturity.1m,type.1m.put,GS.1m.put.strike,round(GS.1m.put.bid,3),
                               round(GS.1m.put.ask,3),round(market.price.1m.put,3),
                               round(explicit.price.1m.put,3),round(implicit.price.1m.put,3),
                               round(crank.nicolson.price.1m.put,3),round(bisec.vol.1m.put,3))
detailed.table.2m.call = cbind(maturity.2m,type.2m.call,GS.2m.call.strike,round(GS.2m.call.bid,3),
                               round(GS.2m.call.ask,3),round(market.price.2m.call,3),
                               round(explicit.price.2m.call,3),round(implicit.price.2m.call,3),
                               round(crank.nicolson.price.2m.call,3),round(bisec.vol.2m.call,3))
detailed.table.2m.put = cbind(maturity.2m,type.2m.put,GS.2m.put.strike,round(GS.2m.put.bid,3),
                               round(GS.2m.put.ask,3),round(market.price.2m.put,3),
                               round(explicit.price.2m.put,3),round(implicit.price.2m.put,3),
                               round(crank.nicolson.price.2m.put,3),round(bisec.vol.2m.put,3))
detailed.table.3m.call = cbind(maturity.3m,type.3m.call,GS.3m.call.strike,round(GS.3m.call.bid,3),
                               round(GS.3m.call.ask,3),round(market.price.3m.call,3),
                               round(explicit.price.3m.call,3),round(implicit.price.3m.call,3),
                               round(crank.nicolson.price.3m.call,3),round(bisec.vol.3m.call,3))
detailed.table.3m.put = cbind(maturity.3m,type.3m.put,GS.3m.put.strike,round(GS.3m.put.bid,3),
                               round(GS.3m.put.ask,3),round(market.price.3m.put,3),
                               round(explicit.price.3m.put,3),round(implicit.price.3m.put,3),
                               round(crank.nicolson.price.3m.put,3),round(bisec.vol.3m.put,3))

full.table = rbind(detailed.table.1m.call,detailed.table.1m.put,detailed.table.2m.call,
                   detailed.table.2m.put,detailed.table.3m.call,detailed.table.3m.put)
colnames(full.table) = c('t(days)','Type','Strike','Bid','Ask','Market','EFD',
                         'IFD','CNFD','Implied Vol')
data.frame.full.table=as.data.frame(full.table)
data.frame.full.table
```

```
##    t(days) Type Strike   Bid   Ask Market    EFD    IFD   CNFD Implied Vol
## 1       17 Call  222.5 30.05    32 31.025 31.037 31.014 31.024       0.894
## 2       17 Call    225 27.55 29.35  28.45 28.443 28.426 28.436       0.841
## 3       17 Call  227.5  25.1 27.15 26.125 26.126 26.114 26.126       0.801
## 4       17 Call    230 22.75  23.4 23.075 23.088 23.037 23.049       0.722
## 5       17 Call  232.5 20.15 22.25   21.2 21.174 21.187   21.2       0.703
## 6       17 Call    235 17.85  18.5 18.175 18.185 18.137 18.151       0.625
## 7       17 Call  237.5  15.4  17.5  16.45 16.451 16.434 16.449        0.61
## 8       17 Call    240  13.3 14.05 13.675 13.672 13.636 13.652       0.542
## 9       17 Call  242.5  11.2  11.8   11.5 11.512 11.481 11.497         0.5
## 10      17 Call    245  9.25  9.75    9.5  9.492   9.46  9.476       0.463
## 11      17  Put  222.5   0.1  0.17  0.135  0.132  0.156  0.152       0.229
## 12      17  Put    225  0.12   0.2   0.16  0.162  0.172  0.168       0.213
## 13      17  Put  227.5  0.11  0.25   0.18  0.178  0.208  0.204       0.193
## 14      17  Put    230   0.2   0.3   0.25  0.252  0.249  0.245       0.181
## 15      17  Put  232.5  0.32  0.38   0.35  0.353  0.382   0.38       0.168
## 16      17  Put    235  0.43   0.5  0.465  0.464  0.454  0.452       0.152
## 17      17  Put  237.5  0.58  0.66   0.62  0.615  0.645  0.645       0.135
## 18      17  Put    240  0.81   0.9  0.855  0.853  0.811  0.812       0.116
## 19      17  Put  242.5  1.15  1.24  1.195    1.2  1.172  1.176       0.094
## 20      17  Put    245  1.61  1.73   1.67  1.667  1.475  1.477       0.062
```

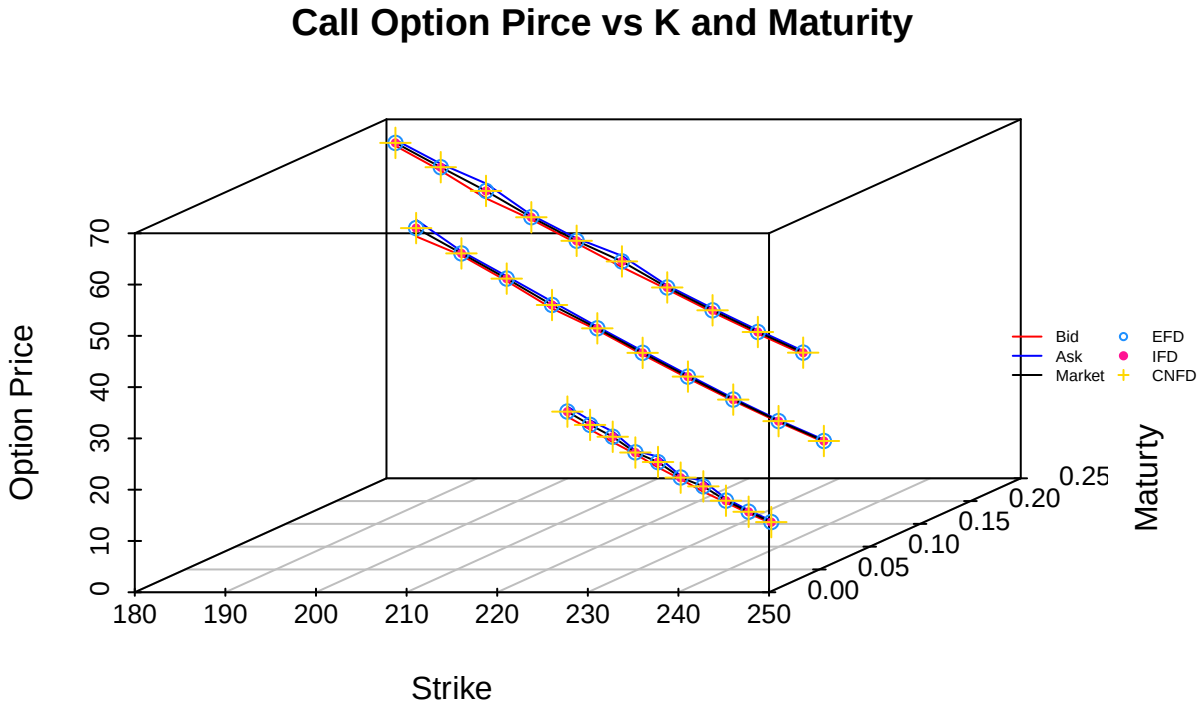
```
## 21      52 Call    195 56.55  59.8 58.175 58.174 58.171 58.173       0.799
## 22      52 Call    200 52.85 53.65  53.25 53.237 53.236 53.239       0.746
## 23      52 Call    205 47.85  48.8 48.325 48.325 48.319 48.323       0.694
## 24      52 Call    210  42.5 43.85 43.175 43.182 43.172 43.177       0.634
## 25      52 Call    215  38.3 38.95 38.625 38.631 38.621 38.626       0.594
## 26      52 Call    220  33.5  34.2  33.85 33.847 33.839 33.845       0.546
## 27      52 Call    225  28.9 29.55 29.225 29.213 29.206 29.213       0.502
## 28      52 Call    230  24.5    25  24.75 24.738 24.718 24.726        0.46
## 29      52 Call    235 20.35 20.75  20.55  20.54 20.521 20.529       0.423
## 30      52 Call    240 16.45  16.9 16.675 16.658 16.648 16.656        0.39
## 31      52  Put    195  0.14  0.25  0.195  0.195  0.203    0.2       0.294
## 32      52  Put    200  0.22  0.28   0.25  0.251  0.254  0.251       0.277
## 33      52  Put    205   0.3  0.37  0.335  0.336  0.345  0.342       0.261
## 34      52  Put    210  0.34   0.5   0.42  0.421  0.433  0.431       0.241
## 35      52  Put    215  0.59  0.66  0.625  0.623  0.638  0.636        0.23
## 36      52  Put    220  0.83   0.9  0.865  0.867  0.872   0.87       0.214
## 37      52  Put    225  1.17  1.28  1.225  1.221  1.217  1.217         0.2
## 38      52  Put    230  1.67  1.81   1.74  1.741  1.705  1.706       0.184
## 39      52  Put    235  2.43  2.58  2.505  2.509  2.465  2.468       0.169
## 40      52  Put    240   3.5  3.65  3.575  3.575  3.528  3.532       0.151
## 41      77 Call    185 68.05  69.2 68.625 68.615 68.618  68.62       0.763
## 42      77 Call    190  63.2 64.55 63.875 63.873 63.874 63.876       0.724
## 43      77 Call    195  57.8  60.7  59.25 59.251 59.245 59.248       0.689
## 44      77 Call    200  53.7 54.55 54.125 54.129 54.111 54.114       0.639
## 45      77 Call    205 49.05    50 49.525 49.528 49.519 49.522       0.604
## 46      77 Call    210  44.4  46.6   45.5 45.502 45.497 45.501       0.586
## 47      77 Call    215 40.05  40.8 40.425 40.417  40.42 40.425       0.536
## 48      77 Call    220  35.6 36.35 35.975 35.964 35.964  35.97       0.503
## 49      77 Call    225  31.4  32.1  31.75 31.744 31.733 31.739       0.474
## 50      77 Call    230  27.4 28.05 27.725 27.719 27.698 27.705       0.448
## 51      77  Put    185  0.55  0.62  0.585  0.587   0.59  0.587        0.35
## 52      77  Put    190  0.67  0.75   0.71  0.712  0.719  0.716       0.334
## 53      77  Put    195  0.86  0.91  0.885  0.887  0.892   0.89       0.321
## 54      77  Put    200  1.06  1.14    1.1  1.103  1.098  1.096       0.307
## 55      77  Put    205  1.36  1.45  1.405  1.407   1.41  1.409       0.295
## 56      77  Put    210  1.74  1.83  1.785  1.782  1.793  1.792       0.283
## 57      77  Put    215  2.22  2.31  2.265  2.264  2.271  2.271       0.272
## 58      77  Put    220  2.84  2.94   2.89  2.894  2.888  2.889        0.26
## 59      77  Put    225   3.6  3.75  3.675  3.673  3.659  3.661       0.249
## 60      77  Put    230  4.55  4.75   4.65  4.654  4.615  4.618       0.237
```

Plot of all Call Option Pice vs Maturity and Strike

In this part, we will use a 3D plot to plot call option price vs maturity and strike.

```
library(scatterplot3d)
full.call.table = rbind(full.table[1:10,],full.table[21:30,],full.table[41:50,])
maturity.1m.year = maturity.1m/360
coordinate.1m.call = cbind(GS.1m.call.strike,GS.1m.call.bid,maturity.1m)

call.3d = scatterplot3d(x=as.numeric(full.call.table[1:10,3]),y = as.numeric(full.call.table[1:10,1])/3
              z = as.numeric(full.call.table[1:10,4]),type = 'l',
              xlim = c(185,245),ylim = c(0,1/4),zlim = c(0,70),color = 'red',
              main = 'Call Option Pirce vs K and Maturity',xlab = 'Strike',
```

```
              ylab = 'Maturty', zlab = 'Option Price')
call.3d$points3d(as.numeric(full.call.table[1:10,3]),as.numeric(full.call.table[1:10,1])/360,
                 as.numeric(full.call.table[1:10,5]),type = 'l',col = 'blue')
call.3d$points3d(as.numeric(full.call.table[1:10,3]),as.numeric(full.call.table[1:10,1])/360,
                 as.numeric(full.call.table[1:10,6]),type = 'l',col = 'black')
call.3d$points3d(as.numeric(full.call.table[1:10,3]),as.numeric(full.call.table[1:10,1])/360,
                 as.numeric(full.call.table[1:10,7]),pch = 21,
                 type = 'p',col = 'dodgerblue')
call.3d$points3d(as.numeric(full.call.table[1:10,3]),as.numeric(full.call.table[1:10,1])/360,
                 as.numeric(full.call.table[1:10,8]),pch = 19,type = 'p',col = 'deeppink',cex = 0.5)
call.3d$points3d(as.numeric(full.call.table[1:10,3]),as.numeric(full.call.table[1:10,1])/360,
                 as.numeric(full.call.table[1:10,9]),pch = 3,type = 'p',col = 'gold',cex = 1.5)

call.3d$points3d(as.numeric(full.call.table[11:20,3]),as.numeric(full.call.table[11:20,1])/360,
                 as.numeric(full.call.table[11:20,4]),type = 'l',col = 'red')
call.3d$points3d(as.numeric(full.call.table[11:20,3]),as.numeric(full.call.table[11:20,1])/360,
                 as.numeric(full.call.table[11:20,5]),type = 'l',col = 'blue')
call.3d$points3d(as.numeric(full.call.table[11:20,3]),as.numeric(full.call.table[11:20,1])/360,
                 as.numeric(full.call.table[11:20,6]),type = 'l',col = 'black')
call.3d$points3d(as.numeric(full.call.table[11:20,3]),as.numeric(full.call.table[11:20,1])/360,
                 as.numeric(full.call.table[11:20,7]),pch = 21,
                 type = 'p',col = 'dodgerblue')
call.3d$points3d(as.numeric(full.call.table[11:20,3]),as.numeric(full.call.table[11:20,1])/360,
                 as.numeric(full.call.table[11:20,8]),pch = 19,type = 'p',col = 'deeppink',cex = 0.5)
call.3d$points3d(as.numeric(full.call.table[11:20,3]),as.numeric(full.call.table[11:20,1])/360,
                 as.numeric(full.call.table[11:20,9]),pch = 3,type = 'p',col = 'gold',cex = 1.5)

call.3d$points3d(as.numeric(full.call.table[21:30,3]),as.numeric(full.call.table[21:30,1])/360,
                 as.numeric(full.call.table[21:30,4]),type = 'l',col = 'red')
call.3d$points3d(as.numeric(full.call.table[21:30,3]),as.numeric(full.call.table[21:30,1])/360,
                 as.numeric(full.call.table[21:30,5]),type = 'l',col = 'blue')
call.3d$points3d(as.numeric(full.call.table[21:30,3]),as.numeric(full.call.table[21:30,1])/360,
                 as.numeric(full.call.table[21:30,6]),type = 'l',col = 'black')
call.3d$points3d(as.numeric(full.call.table[21:30,3]),as.numeric(full.call.table[21:30,1])/360,
                 as.numeric(full.call.table[21:30,7]),pch = 21,
                 type = 'p',col = 'dodgerblue')
call.3d$points3d(as.numeric(full.call.table[21:30,3]),as.numeric(full.call.table[21:30,1])/360,
                 as.numeric(full.call.table[21:30,8]),pch = 19,type = 'p',col = 'deeppink',cex = 0.5)
call.3d$points3d(as.numeric(full.call.table[21:30,3]),as.numeric(full.call.table[21:30,1])/360,
                 as.numeric(full.call.table[21:30,9]),pch = 3,type = 'p',col = 'gold',cex = 1.5)

legend('right',c('Bid','Ask','Market','EFD','IFD','CNFD'),bty = 'n',
               col = c('red','blue','black','dodgerblue','deeppink','gold'),lty = c(1,1,1,NA,NA,NA),
               pch = c(NA,NA,NA,21,19,3),ncol = 2,cex = 0.5,inset = -0.1, xpd = TRUE, horiz = FALSE)
```



Comments on Table and Plot

Based on the table and plot, we can find that for all of the explicit, implicit and crank-nicolson finite difference methods, the result price is very close to the market pirce. And therefore, the points will stay in between bid and ask line.

And when strike increase, the option value will decrease because it is call option. When time goes by, time to maturity decrease, the option value will decrease. It is because stock has less time so that it has less opptunity to volatility. It also follows that theta of call option is negative.