Введение в финансовую математику. Лабораторная работа № 4.

Zhdanov Pavel

07.05.2019 г

Вариант 12

Задание 1

Пусть функция f(X) задана на отрезке [7.9;8.1], и модуль ее второй производной не превосходит 1.2, модуль третьей производной не превосходит 1.3, а модуль четвертой производной не превосходит 1. Найдите оценки на f'(8.0) и f''(8.0) и теоретические погрешности этих оценок, если:

f(7.98) = 3.069007

f(8) = 3.0688

f(8.02) = 3.06819

Сравнить результаты с точными, считая, что

$$f(x) = \frac{x^3}{6} - 4x^2 + 32x + ln(x) + sin(x) - \frac{256}{3}$$

Решение

Численные производные:
1. Правосторонняя:

$$f'_r(x) = \frac{f(x+h) - f(x)}{h} \sim f'(x) + \frac{f''(x^*)h}{2!}$$

2. Левосторонняя:

$$f'_l(x) = \frac{f(x) - f(x-h)}{h} \sim f'(x) + \frac{f''(x^*)h}{2!}$$

3. Симметричная:

$$f'_c(x) = \frac{f(x+h) - f(x-h)}{2h} \sim f'(x) + \frac{f'''(x^*)h^2}{3!}$$

Объявим переменные

```
f_left_meaning = 3.069007
f_meaning = 3.0688
f_right_meaning = 3.06819
h = 0.02
```

Считаем: 1. Правостороняя:

```
f_r = (f_right_meaning - f_meaning)/h
print(f_r)

## [1] -0.0305
```

2. Левостороняя:

```
f_l = (f_meaning - f_left_meaning)/h
print(f_l)

## [1] -0.01035
```

3. Симметричная

```
f_c = (f_right_meaning - f_left_meaning)/(2*h)
print(f_c)

## [1] -0.020425
```

Производная исходной функции:

```
derivative <- function(x){
  ans = (x * x) / 2 - 8 * x + 32 + 1/x + cos(x)
  return (ans)
}

certain = derivative(8)
print(certain)

## [1] -0.02050003
```

Погрешности аппроксимации:

```
print(f_r - certain)

## [1] -0.009999966

print(f_l - certain)

## [1] 0.01015003

print(f_c - certain)

## [1] 7.503381e-05
```

Вывод: наибольший порядок точности достигается при симметричной аппроксимации, что подтверждается теоретическими выкладками и математическим подсчетом.

Вторая производная:

$$f''(x) = \lim_{h\to 0}\frac{f'(x+h) - f'(x)}{h} = \lim_{h\to 0}\frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

Аппроксимация для второй производной:

$$f''_c(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \sim f''(x) + \frac{f''''(x^*)h^2}{12}$$

```
f_cc = (f_left_meaning - 2 * f_meaning + f_right_meaning)/(h * h)
print(f_cc)

## [1] -1.0075
```

Вторая производная исходной функции

```
deriviate_2 <- function(x) {
  ans = x - 8 - 1/(x^2) - sin(x)
  return (ans)
}

certain = deriviate_2(8)
print(certain)

## [1] -1.004983
```

Погрешность аппроксимации:

```
print(f_cc - certain)

## [1] -0.002516753
```

Задание 2

Реализовать явную схему для подсчета цен Европейских и Американских опционов. Шаги:
* Выбрать подходящее количество шагов I, для устойчивости
* Посчитать граничные и финальные состояния
* Посчитать pu, pm, pd

Построить графики зависимости полученных из явной схемы цен от K, используя следующие параметры: s_call = 17$, s_put = 14$, T = 3, r = 7%, sigma = 30%, K = [30, 60, ... ,270]. Сравнить полученные результаты :
* Для Европейских опционов (S = S_call для call и S = S_put для put)) - с Black_Scholes.
* Для Американского Put (S = S_put) - с Биномиальной моделью с 2000 шагов

Решение

Объявление констант

```
S_call = 17
S_put = 14
X = 15
T = 3
r = 0.07
sig = 0.3
S_min = 0
counter = 4
```

Payoff

```
call_payoff_func <- function(spot_price){
  if (spot_price > X){
    return (spot_price - X)
  }
  return (0)
}

put_payoff_func <- function(spot_price){
  if (spot_price < X){
    return (X - spot_price)
  }
  return (0)
}
```

Цены европейских опционов(аналитическое решение)

```
d1 <- function(S){
  ans = (1/(sig * sqrt(T))) * (log(S/X) + (r + (sig^2)/2) * T)
  return (ans)
}

d2 <- function(S){
  ans = d1(S) - sig * sqrt(T)
  return (ans)
}


call_BS_func <- function(spot_price){
  ans = pnorm(d1(spot_price)) * spot_price - pnorm(d2(spot_price)) * X * exp(-r*
T)
  return (ans)
}

put_BS_func <- function(spot_price){
  ans = pnorm(-d2(spot_price)) * X * exp(-r * T) - pnorm(-d1(spot_price)) * spot
_price
  return (ans)
}
```

Решение для Европейских Call опционов Численное решение

```
K_arr = seq(200, 2000, by = 200)
ans_call_european = rep(0.0, length(K_arr))

for (ind in 1:length(K_arr)){
  K = K_arr[ind]
  S_max = counter * S_call
  I = floor(sqrt(K/(sig * sig * T)))
  I = (I%/%counter) * counter
  delta_S = (S_max - S_min)/I
  delta_T = (T/K)
  v_cur = rep(0.0, K)
  v_prev = rep(0.0, K)
  v_app = matrix(rep(0.0, K + 1), (K + 1) * (I + 1), nrow = I + 1, ncol = K + 1)
  v_call_prices = rep(0.0, I + 1)

  for (i in 1:(I + 1)){
    v_app[i, K + 1] = call_payoff_func(S_min + delta_S * (i - 1))
    v_call_prices[i] = S_min + delta_S * (i -1)
  }

  for (k in 1:(K + 1)){
    v_app[1, k] = 0
  }

  for (k in 1:(K + 1)){
    v_app[I + 1, k] = S_max - X * exp(-r * (T - delta_T * (k - 1)))
  }


  for (k in ((K+1):2)){
    for (i in (1:(I-1))){
      p_u = ((sig ^ 2) * (i ^ 2) + r * i) * delta_T/2.0
      p_m = 1 - (sig ^ 2) * (i ^ 2) * delta_T
      p_d = ((sig ^ 2) * (i ^ 2) - r * i) * delta_T/2.0
      v_app[i + 1, k - 1] =  (p_d * v_app[i, k] + p_m * v_app[i + 1, k] + p_u *
v_app[i + 2, k]) / (1.0 + r * delta_T)
    }
  }
  ans_call_european[ind] = v_app[length(v_call_prices)/counter + 1, 1]

}
```

Точное решение для Европейских Call опционов

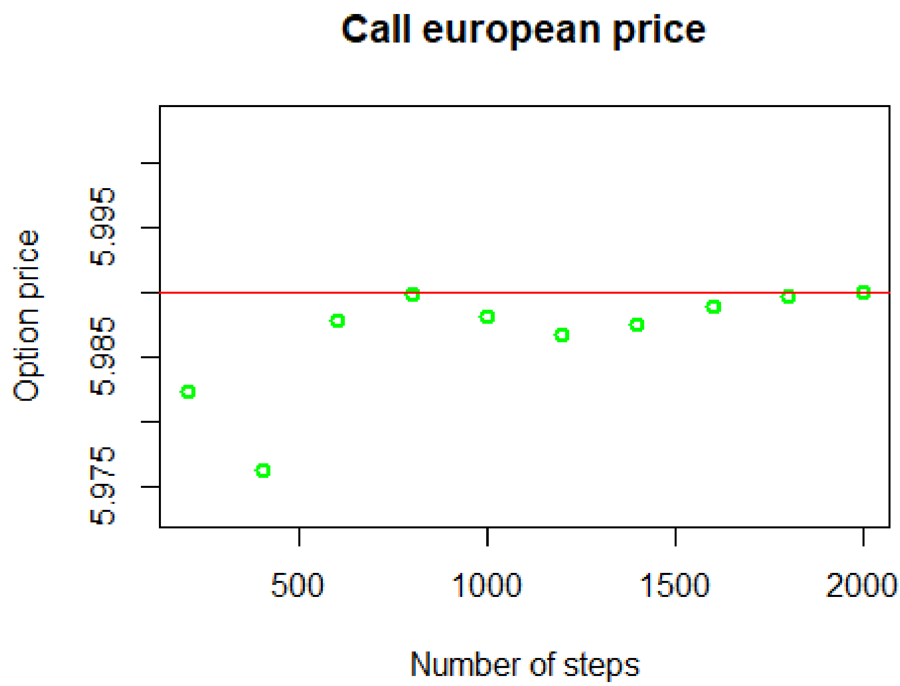
```
certain_solution_call_european = call_BS_func(S_call)
print(certain_solution_call_european)

## [1] 5.990058
```

График с точным и с численым решением для Европейских Call опционов

```
y_start = ans_call_european[5] - 0.015
y_finish = ans_call_european[5] + 0.015
```

```
inds = seq(1, length(ans_call_european))
plot(x = K_arr, y = ans_call_european, col=rgb(0.0,1.0,0.0,1.0), ylim = c(y_start, y_finish),
     lwd=2.0, type = 'p', ylab = 'Option price', xlab='Number of steps', main='Call european price')
abline(h = certain_solution_call_european, col='red')
```



Решение для Европейских Put опционов

```
ans_put_european = rep(0.0, length(K_arr))

for (ind in 1:length(K_arr)){
  K = K_arr[ind]
  S_max = counter * S_put
  I = floor(sqrt(K/(sig * sig * T)))
  I = I %/% counter * counter
  delta_S = (S_max - S_min)/I
  delta_T = (T/K)
  v_cur = rep(0.0, K)
  v_prev = rep(0.0, K)

  v_app = matrix(rep(0.0, K + 1), (K + 1) * (I + 1), nrow = I + 1, ncol = K + 1)
  v_put_prices = rep(0.0, I + 1)

  for (i in 1:(I + 1)){
    v_app[i, K + 1] = put_payoff_func(S_min + delta_S * (i - 1))
    v_put_prices[i] = S_min + delta_S * (i -1)
  }
```

```
  for (k in 1:(K + 1)){
    v_app[1, k] = X * exp(-r * (T - delta_T * (k - 1)))
  }

  for (k in 1:(K + 1)){
    v_app[I + 1, k] = 0
  }

  for (k in ((K+1):2)){
    for (i in (1:(I-1))){
      p_u = ((sig ^ 2) * (i ^ 2) + r * i) * delta_T/2.0
      p_m = 1 - (sig ^ 2) * (i ^ 2) * delta_T
      p_d = ((sig ^ 2) * (i ^ 2) - r * i) * delta_T/2.0
      v_app[i + 1, k - 1] =  (p_d * v_app[i, k] + p_m * v_app[i + 1, k] + p_u *
v_app[i + 2, k]) / (1.0 + r * delta_T)
    }
  }
  ans_put_european[ind] = v_app[length(v_put_prices)/counter + 1, 1]
}
```

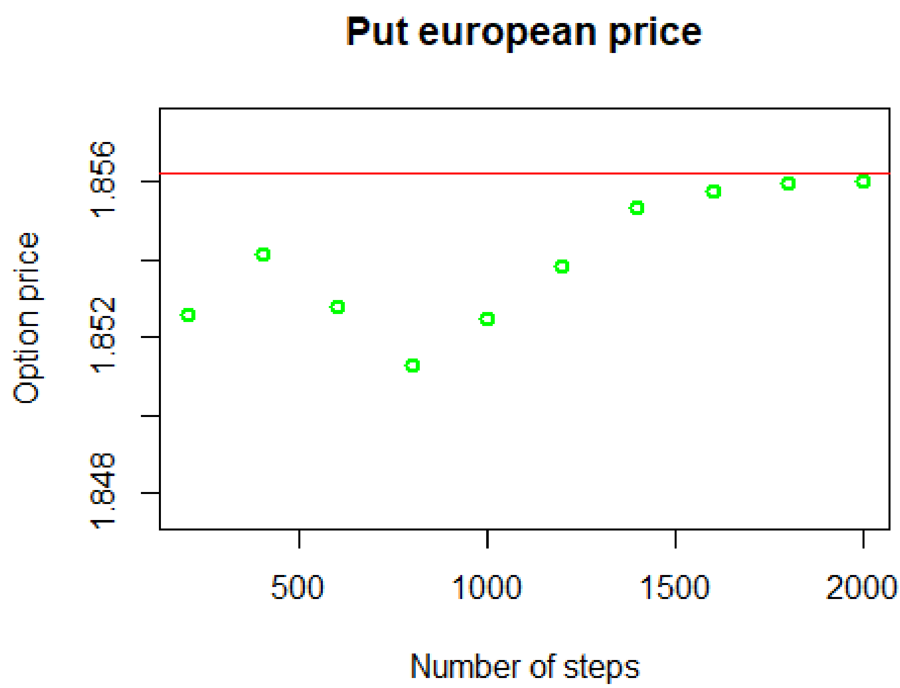
Точное решение для Европейских Put опционов

```
certain_solution_put_european = put_BS_func(S_put)
print(certain_solution_put_european)

## [1] 1.856228
```

График с точным и с численыv решением для Европейских Put опционов

```
y_start = ans_put_european[5] - 0.005
y_finish = ans_put_european[5] + 0.005
inds = seq(1, length(ans_put_european))
plot(x = K_arr, y = ans_put_european, col=rgb(0.0,1.0,0.0,1.0), ylim = c(y_start
, y_finish),
     lwd=2.0, type = 'p', ylab = 'Option price', xlab='Number of steps', main='P
ut european price')
abline(h = certain_solution_put_european, col='red')
```



Решение для Американских Put опционов

Численное решение

```
ans_put_american = rep(0.0, length(K_arr))

for (ind in 1:length(K_arr)){
  K = K_arr[ind]
  S_max = counter * S_put
  I = floor(sqrt(K/(sig * sig * T)))
  I = I%/%counter * counter
  delta_S = (S_max - S_min)/I
  delta_T = (T/K)
  v_cur = rep(0.0, K)
  v_prev = rep(0.0, K)

  v_app = matrix(rep(0.0, K + 1), (K + 1) * (I + 1), nrow = I + 1, ncol = K + 1)
  v_put_prices = rep(0.0, I + 1)

  for (i in 1:(I + 1)){
    v_app[i, K + 1] = put_payoff_func(S_min + delta_S * (i - 1))
    v_put_prices[i] = S_min + delta_S * (i-1)
  }

  for (k in 1:(K + 1)){
    v_app[1, k] = X
  }

  for (k in 1:(K + 1)){
```

```
    v_app[I + 1, k] = 0
  }

  for (k in ((K+1):2)){
    for (i in (1:(I-1))){
      p_u = ((sig ^ 2) * (i ^ 2) + r * i) * delta_T/2.0
      p_m = 1 - (sig ^ 2) * (i ^ 2) * delta_T
      p_d = ((sig ^ 2) * (i ^ 2) - r * i) * delta_T/2.0
      v_app[i + 1, k - 1] =  max((p_d * v_app[i, k] + p_m * v_app[i + 1, k] + p_
u * v_app[i + 2, k]) / (1.0 + r * delta_T), put_payoff_func(v_put_prices[i+1]))
    }
  }
  ans_put_american[ind] = v_app[length(v_put_prices)/counter + 1, 1]
}
```

Численное решение с помощью биномиальной модели

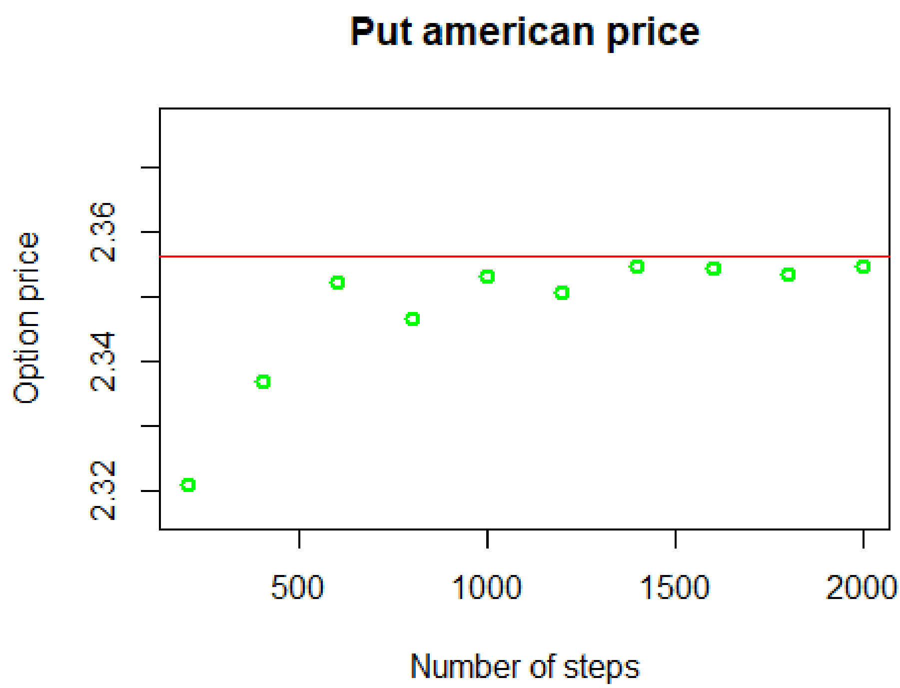
```
library('fOptions')

certain_solution_put_american = CRRBinomialTreeOption(TypeFlag = c("pa"), S_put,
X, T, r, r, sig, 2000)@price
print(certain_solution_put_american)

## [1] 2.356249
```

График с биномиальной моделью и с численыv решением(явная разностная схема) для Американскх Put опционов

```
y_start = ans_put_american[4] - 0.03
y_finish = ans_put_american[4] + 0.03
inds = seq(1, length(ans_put_american))
plot(x = K_arr, y = ans_put_american, col=rgb(0.0,1.0,0.0,1.0), ylim = c(y_start
, y_finish),
     lwd=2.0, type = 'p', ylab = 'Option price', xlab='Number of steps', main='P
ut american price')
abline(h = certain_solution_put_american, col='red')
```



Выводы.

В данной работе рассматривалась разностная схема для приближенного нахождения цен Европейских и Американских опционов. Суть метода состоит в замене производной ее разностным аналогом. Схема может быть записана в виде следующего алгоритма:
1. Задаем количество шагов I для устойчивости:

$$K = \sigma^2 I^2 T$$

2. Считаем граничные и финальные состояния.
3.

$$p_u = \frac{(\sigma^2 i^2 + ri)\Delta t}{2}; p_m = 1 - \sigma^2 i^2; p_d = \frac{(\sigma^2 i^2 - ri)\Delta t}{2}$$

4. Из состояний на уровне k высчитываем состояния на уровне k + 1 по формуле

$$V_i^{k+1} = \frac{p_d V_{i-1}^k - p_m V_i^k + p_u V_{i+1}^k}{1 + r\Delta t}$$

Исходя из построенных графиков, можно прийти к выводу, что решения полученные с помощью явной разностной схемы сходятся к точным решениям(в случае для Американских Put опционов для биномиальной модели). Если необходимо посчитать решение для европейских опционов, то лучше воспользоваться аналитическим решением, поскольку численное решение менее точное и вычислительно трудозатратное. В случае с Американскими Put опционами, если необходимо быстрое решение, то лучше использовать явную разностную схему. Если же нужно более точное решение, то лучше использовать биномиальную модель с большим числом шагов.