Wstęp do matematyki finansów
Sprawozdanie 1

Justyna Niedźwiedzka 229877

11 grudnia 2020

Zadanie 1

W celu wyceny opcji europejskich na drzewku dwumianowym będziemy korzystać z zaimplementowanych funkcji. Bedziemy rozpatrywać przypadki konstrukcji drzewka metodą CRR oraz JR. Parametrami naszych funkcji będą:

- S - cena początkowa,
- K - cena wykonania opcji,
- T - termin wykonania opcji,
- r - stopa procentowa (ciągła),
- σ - zmienność rynku,
- call/put - opcja kupna/ sprzedaży,
- n - liczba kroków na drzewku.

Zbadamy także zbieżność ceny opcji wyznaczonej metodą CRR oraz JR w zależności od parametru n do ceny Blacka-Scholesa.

Cena Blacka-Scholesa europejskich opcji kupna (C) oraz sprzedaży (P) jest dana przez

$$C = S_0\Phi(d_1) - Ke^{-rT}\Phi(d_2),$$
$$P = -S_0\Phi(-d_1) + Ke^{-rT}\Phi(-d_2),$$

gdzie

$$d_1 = \frac{log\frac{S_0}{K} + (r + \frac{\sigma^2}{2})T}{\sqrt{T}\sigma},$$
$$d_2 = \frac{log\frac{S_0}{K} + (r - \frac{\sigma^2}{2})T}{\sqrt{T}\sigma}.$$

```
BS_price <- function(S, K, T, r, sigma, type=c("call","put")){
  d1 = (log(S/K) + (r + (sigma^2)/2) * T) / sqrt(T)*sigma
  d2 = (log(S/K) + (r - (sigma^2)/2) * T) / sqrt(T)*sigma
  if (type=="call"){
    call = S*pnorm(d1) - K*exp(-r*T)*pnorm(d2)
    return(call)
  }
  if (type=="put"){
    put = -S*pnorm(-d1) + K*exp(-r*T)*pnorm(-d2)
    return(put)
  }
}
```

W funkcjach korzystających z metod CRR oraz JR będziemy używać poniższych wzorów, aby wyznaczyć cenę opcji kupna i sprzedaży:

$$C = exp(-rT)\sum_{k=0}^{n}\binom{n}{k}p^k(1-p)^{n-k}\, max(Su^kd^{n-k}-K,0),$$
$$P = exp(-rT)\sum_{k=0}^{n}\binom{n}{k}p^k(1-p)^{n-k}\, max(K-Su^kd^{n-k},0).$$

Zamiast funkcji generującej całe drzewko korzystamy z tego sposobu, ponieważ jest on wygodniejszy w przypadku dużych n, a także wymaga mniej obliczeń.

Poniżej przedstawiono zaimplementowane funkcje wyznaczające cenę opcji kupna i sprzedaży wymienionymi wcześniej metodami. W obu funkcjach korzystamy z przybliżeń dla współczynników u i d.

```
CRR_approx <- function(S, K, T, r, sigma, n, type = c("call", "put")){
  dt <- T/n
  u <- exp(sigma * sqrt(dt))
  d <- 1/u
  p <- (exp(r*dt) - d)/(u - d)
  Df <- exp(-r*T)
  sum <- 0
  if(type == "call"){
    for (k in 0:n){
      sum <- sum + max(S*(u^k)*(d^(n-k)) - K, 0) * dbinom(k, n, prob = p)
    }
    call <- sum*Df
    return(call)
  }
  if(type == "put"){
    for (k in 0:n){
      sum <- sum + max(K - S*(u^k)*(d^(n-k)), 0) * dbinom(k, n, prob = p)
    }
    put <- sum*Df
    return(put)
  }
}
```

```
JR_approx <- function(S, K, T, r, sigma, n, type = c("call", "put")){
  dt <- T/n
  u <- exp((r - 0.5*sigma^2)*dt + sigma*sqrt(dt))
  d <- exp((r - 0.5*sigma^2)*dt - sigma*sqrt(dt))
  p <- 1/2
  Df <- exp(-r*T)
  sum <- 0
  if(type == "call"){
    for (k in 0:n){
      sum <- sum + max(S*(u^k)*(d^(n-k)) - K, 0) * dbinom(k, n, prob = p)
```

```
    }
    call <- sum*Df
    return(call)
  }
  if(type == "put"){
    for (k in 0:n){
      sum <- sum + max(K - S*(u^k)*(d^(n-k)), 0) * dbinom(k, n, prob = p)
    }
    put <- sum*Df
    return(put)
  }
}
```

W poniższych tabelach przedstawiono odpowiednio ceny opcji kupna oraz sprzedaży dla parametrów $S = 100$, $K = 100$, $T = 1$, $r = 0.05$, $\sigma = 0.1$ w zależności od metody i liczby kroków n na drzewku dwumianowym.

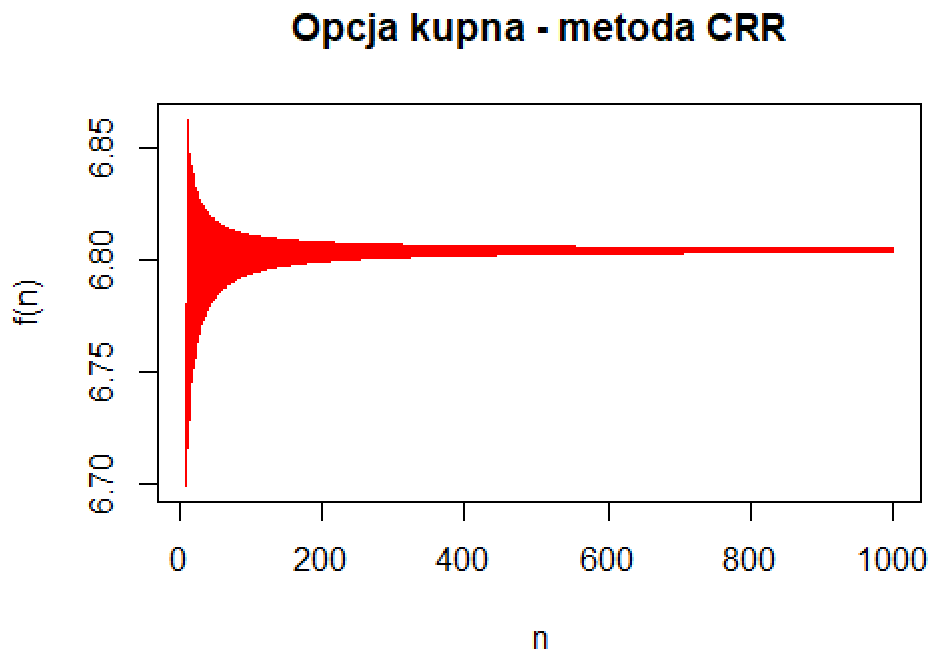
Tabela 1: Ceny opcji kupna

metoda	liczba kroków	cena	czas obliczeń
B-S		6.804958	
CRR	10	6.699215	0.01 s
CRR	100	6.794247	0.02 s
CRR	1000	6.803885	0.05 s
CRR	10000	6.804850	0.14 s
CRR	100000	6.804947	0.47 s
JR	10	6.861830	0.01 s
JR	100	6.808552	0.02 s
JR	1000	6.804745	0.03 s
JR	10000	6.805039	0.11 s
JR	100000	6.804957	0.52 s

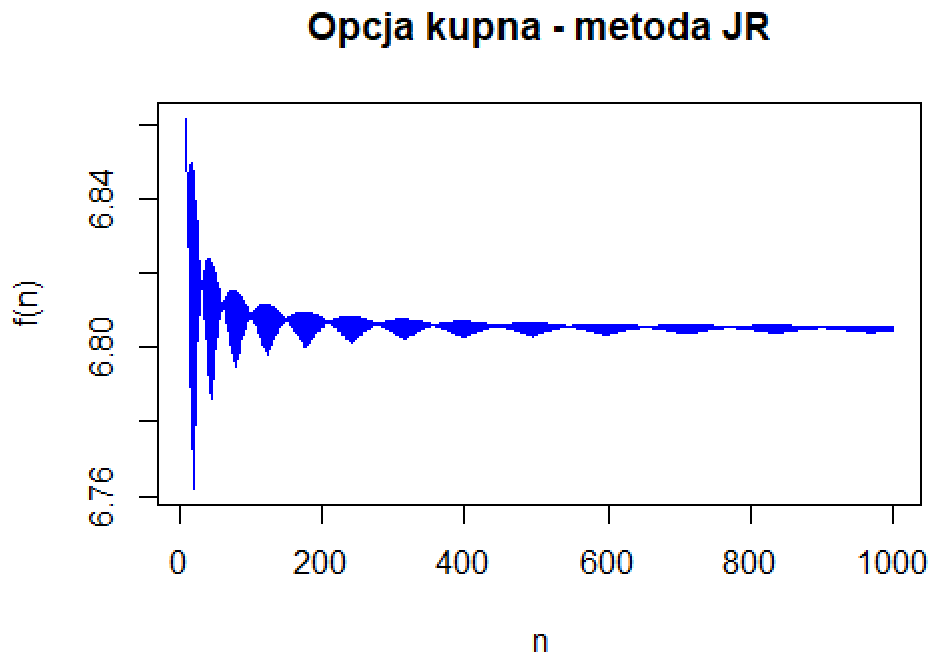
Tabela 2: Ceny opcji sprzedaży

metoda	liczba kroków	cena	czas obliczeń
B-S		1.927900	
CRR	10	1.822158	0.01 s
CRR	100	1.917190	0.02 s
CRR	1000	1.926828	0.03 s
CRR	10000	1.927793	0.11 s
CRR	100000	1.927889	0.47 s
JR	10	1.984856	0.01 s
JR	100	1.931502	0.02 s
JR	1000	1.927688	0.03 s
JR	10000	1.927981	0.09 s
JR	100000	1.927900	0.47 s

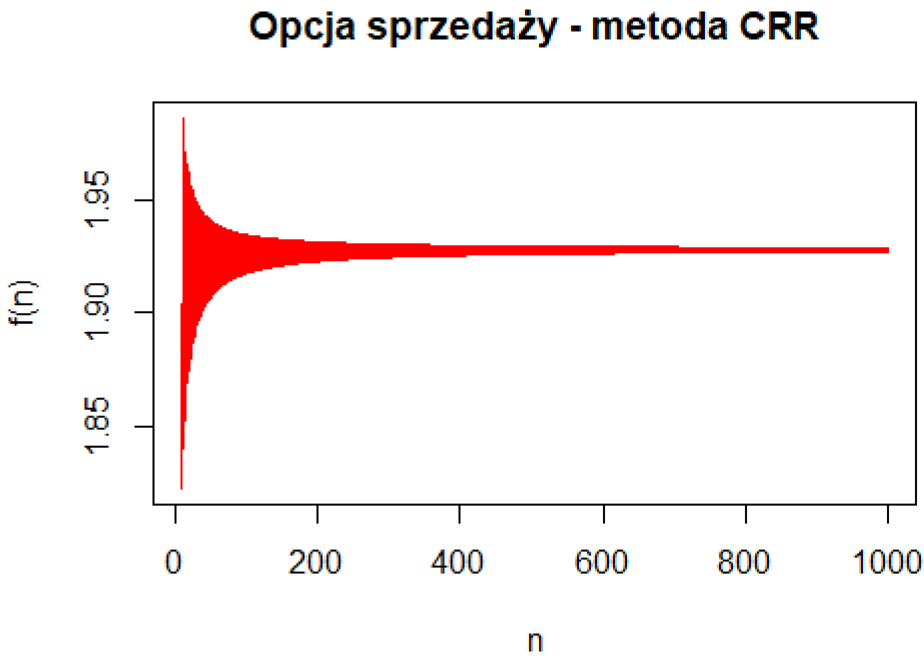
Teraz przedstawimy na kilku wykresach cenę opcji w zależności od ilości kroków n na drzewku dwumianowym.



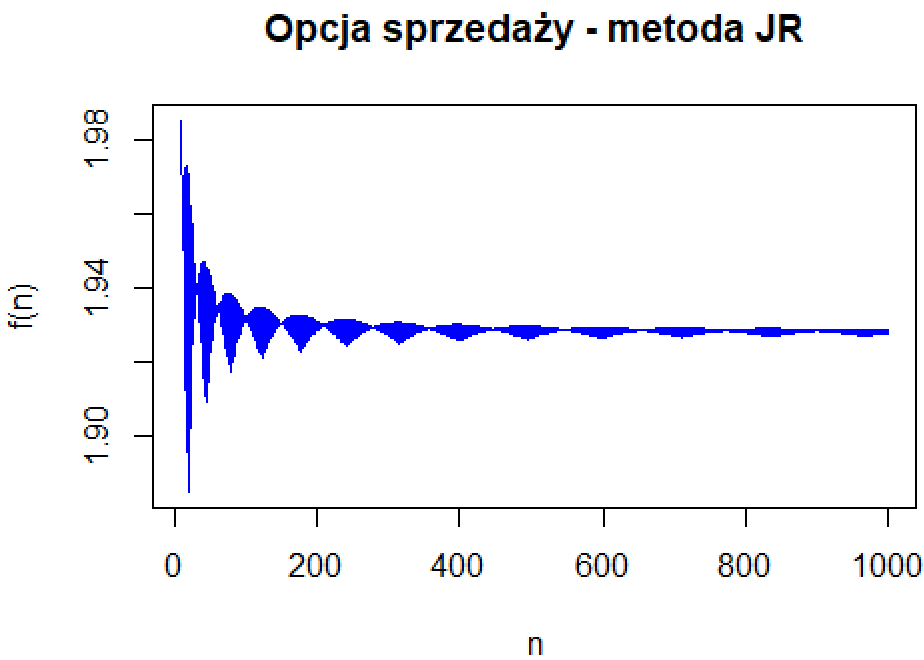
Rysunek 1: Cena opcji sprzedaży, metoda CRR



Rysunek 2: Cena opcji sprzedaży, metoda JR



Rysunek 3: Cena opcji kupna, metoda CRR



Rysunek 4: Cena opcji kupna, metoda JR

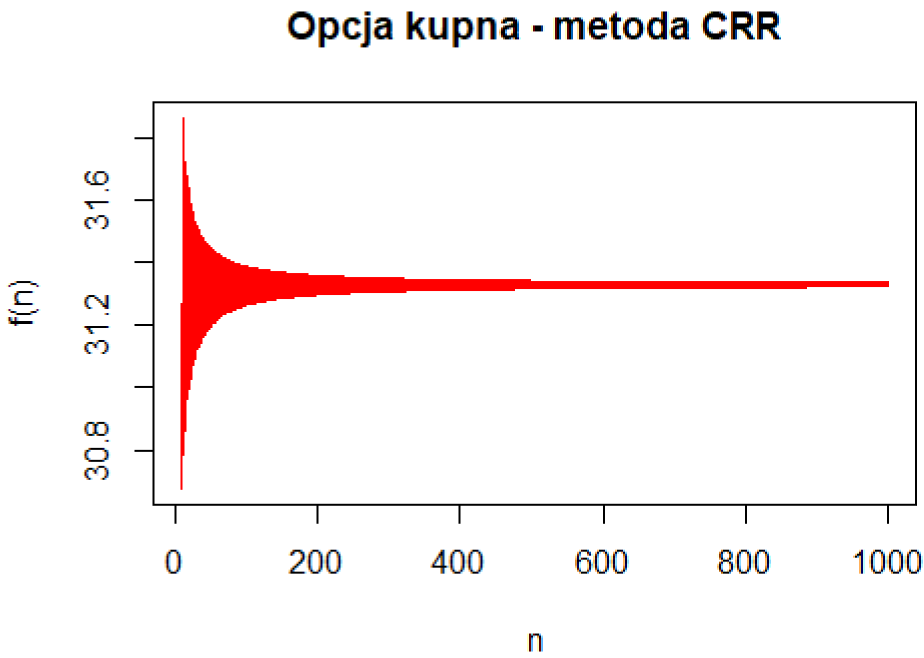
Porównamy jeszcze wyniki dla parametrów $S = 100$, $K = 100$, $T = 2$, $r = 0.05$, $\sigma = 0.5$.

Tabela 3: Ceny opcji kupna

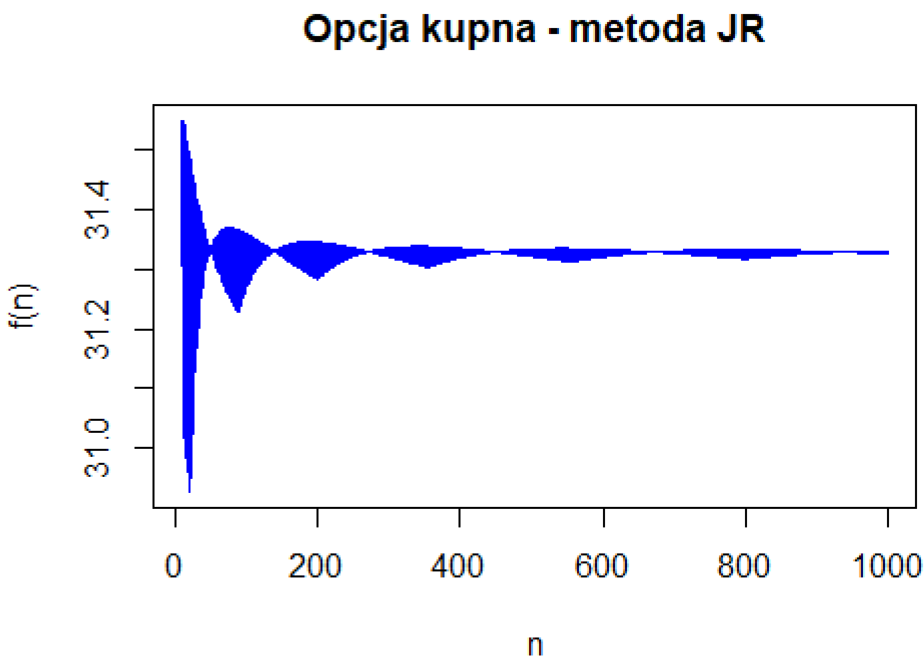
metoda	liczba kroków	cena	czas obliczeń
B-S		31.32768	
CRR	10	30.67453	0.01 s
CRR	100	31.26153	0.02 s
CRR	1000	31.32106	0.03 s
CRR	10000	31.32702	0.12 s
CRR	100000	31.32762	0.61 s
JR	10	31.54712	0.01 s
JR	100	31.26519	0.02 s
JR	1000	31.32999	0.02 s
JR	10000	31.32796	0.08 s
JR	100000	31.32772	0.50 s

Tabela 4: Ceny opcji sprzedaży

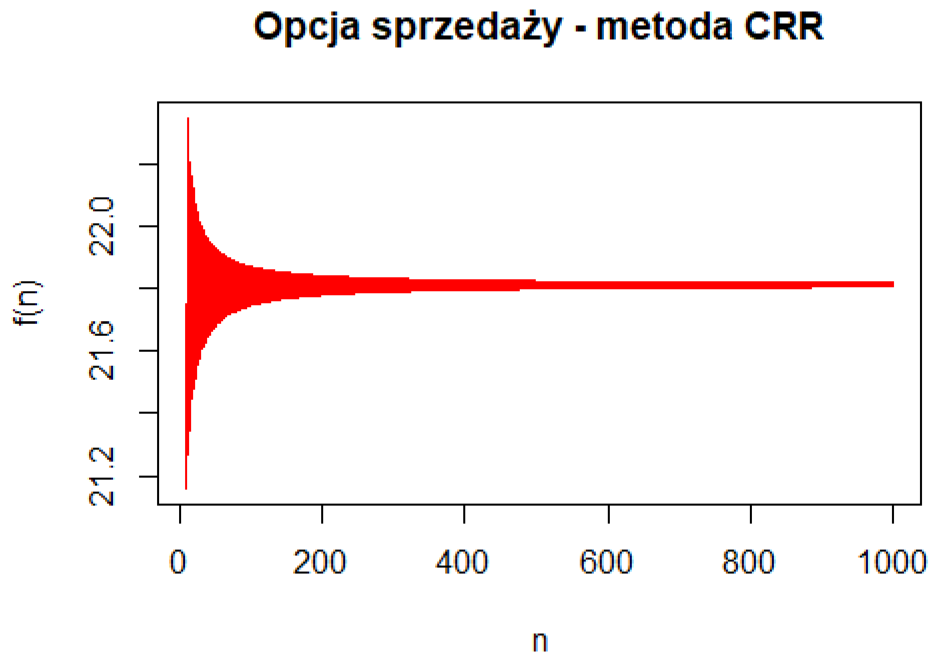
metoda	liczba kroków	cena	czas obliczeń
B-S		21.81143	
CRR	10	21.15828	0.01 s
CRR	100	21.74527	0.01 s
CRR	1000	21.80480	0.02 s
CRR	10000	21.81076	0.08 s
CRR	100000	21.81136	0.49 s
JR	10	22.23625	0.01 s
JR	100	21.76973	0.01 s
JR	1000	21.81581	0.02 s
JR	10000	21.81191	0.08 s
JR	100000	21.81148	0.50 s



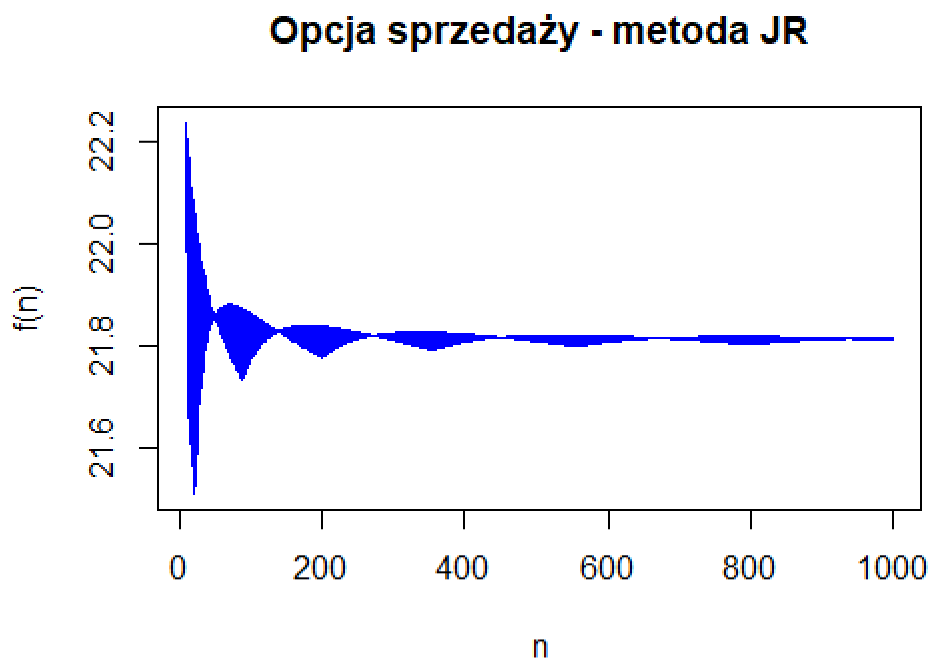
Rysunek 5: Cena opcji sprzedaży, metoda CRR



Rysunek 6: Cena opcji sprzedaży, metoda JR



Rysunek 7: Cena opcji kupna, metoda CRR



Rysunek 8: Cena opcji kupna, metoda JR

Zarówno w przypadku metody CRR jak i JR występuje zbieżność do ceny Blacka-Scholesa. Z wykresów zauważamy, że dla metody JR występują wahania ceny. Biorąc pod uwagę $n = 100000$ możemy stwierdzić, iż metoda JR wypada nieco lepiej od CRR.

Zadanie 2

W tym zadaniu rozważamy obie metody z Zadania 1, biorąc dokładne wyliczenia współczynników u i d. Skorzystamy z zaimplementowanych funkcji:

```
CRR_exact <- function(S, K, T, r, sigma, n, type = c("call", "put")){
  dt <- T/n
  a <- exp(r*dt + dt*sigma^2) + exp(-r*dt)
  u <- 0.5 * (a + sqrt(a^2 - 4))
  d <- 0.5 * (a - sqrt(a^2 - 4))
  p <- (exp(r*dt) - d)/(u - d)
  Df <- exp(-r*T)
  sum <- 0
  if(type == "call"){
    for (k in 0:n){
      sum <- sum + max(S*(u^k)*(d^(n-k)) - K,0) * dbinom(k, n, prob = p)
    }
    call <- sum*Df
    return(call)
  }
  if(type == "put"){
    for (k in 0:n){
      sum = sum + max(K - S*(u^k)*(d^(n-k)),0) * dbinom(k, n, prob = p)
    }
    put <- sum*Df
    return(put)
  }
}
```

```
JR_exact <- function(S, K, T, r, sigma, n, type = c("call", "put")){
  dt <- T/n
  u <- exp(r*dt) * (1 + sqrt(exp(dt*sigma^2) - 1))
  d <- exp(r*dt) * (1 - sqrt(exp(dt*sigma^2) - 1))
  p <- 1/2
  Df <- exp(-r*T)
  sum <- 0
  if(type == "call"){
    for (k in 0:n){
      sum <- sum + max(S*(u^k)*(d^(n-k)) - K, 0) * dbinom(k, n, prob = p)
    }
    call <- sum*Df
    return(call)
  }
  if(type == "put"){
    for (k in 0:n){
      sum <- sum + max(K - S*(u^k)*(d^(n-k)), 0) * dbinom(k, n, prob = p)
    }
```

```
    put <- sum*Df
    return(put)
  }
}
```

Przeprowadzimy analizę dla parametrów $S = 100$, $K = 100$, $T = 1$, $r = 0.05$, $\sigma = 0.1$, aby móc porównać wyniki do tych z poprzedniego zadania.

Tabela 5: Ceny opcji kupna

metoda	liczba kroków	cena	czas obliczeń
B-S		6.804958	
CRR	10	6.749737	0.01 s
CRR	100	6.799446	0.02 s
CRR	1000	6.804407	0.02 s
CRR	10000	6.804903	0.09 s
CRR	100000	6.804952	0.58 s
JR	10	6.864074	0.01 s
JR	100	6.808753	0.02 s
JR	1000	6.804765	0.02 s
JR	10000	6.805041	0.09 s
JR	100000	6.804958	0.55 s

Tabela 6: Ceny opcji sprzedaży

metoda	liczba kroków	cena	czas obliczeń
B-S		1.927900	
CRR	10	1.872679	0.01 s
CRR	100	1.922389	0.01 s
CRR	1000	1.927349	0.02 s
CRR	10000	1.927845	0.08 s
CRR	100000	1.927895	0.45 s
JR	10	1.987016	0.01 s
JR	100	1.931695	0.01 s
JR	1000	1.927708	0.02 s
JR	10000	1.927983	0.07 s
JR	100000	1.927900	0.46 s

Po przeanalizowaniu wyników kolejny raz okazuje się, że metoda JR daje lepsze rezultaty. Dla $n = 100000$ dostajemy dokładnie cenę Blacka-Scholesa. Porównując wyniki do tych z Zadania 1 możemy stwierdzić, że ceny opcji wyznaczonych za pomocą dokładnych wyliczeń współczynników są bliższe cenie B-S. Czasy obliczeń dla każdej z metod są podobne.